ALIEN-NINJA ROV

company **RoboCenter**

school **The Center for Robotics Development**

location **Vladivostok | Primorye Territory | Russia**

Students

Anton Konstantinov, CEO, CFO, Grade 10
Nikolai Dolzhenko, CIO, COO, Grade 10
Marat Rasulov, CTO, Grade 10
Lev Davydenko, Safety Manager, Grade 10
Daniil Proshin, Programmer, Grade 8
Iaroslav Proshin, Electronic engineer, Grade 8
Anton Iakovivskii, Props constructor, Grade 8
Matvei Kostylev, CMO, Grade 10

Mentors Sergey Mun Angelina Borovskaia Denis Mikhailov

Contents

Abstract

Our company RoboCenter is happy to present our ROV Alien-Ninja to participate in MATE competition. Unique in its kind our vehicle is capable of performing lots of underwater operations and of successful solving all the mission tasks. The ROV features multitask capability and reliability. Space effective layout of components provides for tailoring it to suit the clients' needs, while powerful thrusters, a manipulator and four cameras allow for ample room for underwater operations.

The ROV power-feeding voltage is 12V, power demand is 300W, max vertical thrust is 2,6kgf, and horizontal thrust is 3,7kgf. The vehicle is capable of attaining a speed of up to 0,5m/s, proceeding along the set course, self-trimming when in water. One can find a detailed report on the design of our robot and able of attaining a speed of up to 0,8m/s, proceeding along the set course, self-trimming the procedure of preparations for ROV Competition. Final dimensions of the vehicle are: length 600mm, height 332mm, width 420mm.

Fig.1 RoboCenter LLC. Left to right: Daniil Proshin, Nikolai Dolzhenko, Anton Iakovivskii, Anton Konstantinov, Lev Davydenko, Marat Rasulov, Iaroslav Proshin, Matvei Kostylev

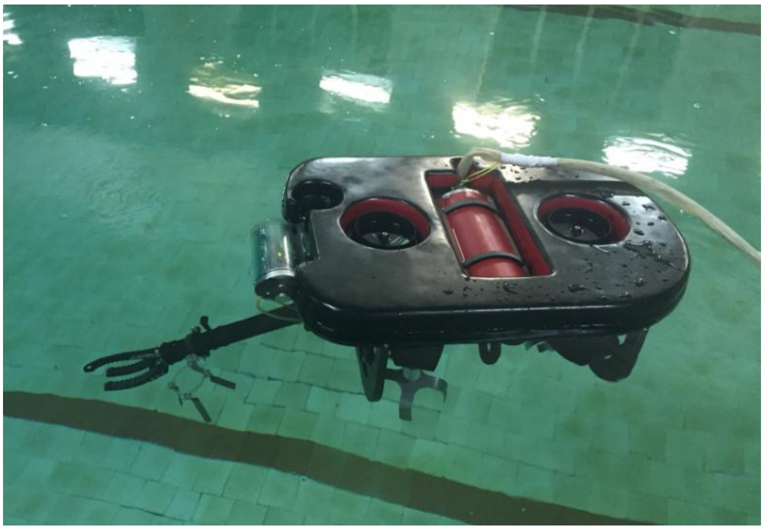


Fig.2 Alien-Ninja ROV

Budget

Money was the first thing we had to think of when going ahead with the project. But our mentors helped us with this problem. They found sponsors and invested the capital of the Center for Development of Robotics (CDR). Having done so the mentors demanded from us accurate book-keeping and cost accounting. All the slips were to be kept with the CFO and he was to enter all the expenditures into a special table and to classify them.

See below our company's expenditures for the project.

Parts and Materials

Item	**Donations, USD**	**Expenditures, USD**	**Provider**
		ROV	
Surface equipment			
Joystick		60	DNS
TV-monitor		100	Technopoint
Pelican case		56	Farpost
Router		40	DNS
Mechanics			
Polycarbonate sheet		220	SMiT
Acrylic, Aluminum		180	Chiner, Megamet
Materials for props		100	DZL
O-rings		40	Rezinotehnika
Fasteners		56	DZL
Other materials	100		CDR
Electronics			
Arduino Uno, Mega		36	TerraElectronics
DC-DC converters		60	TerraElectronics
Pololu dual vnh501 ($60x4)		240	TerraElectronics
LN298N		5	DalCon
Wires		70	Chip&Dip
Connectors		100	Chip&Dip
Arduino Ethernet Shield		14	DalCon
Sensors (pressure, GY-521)		20	DalCon
Pump	60		CDR
Other parts	140		CDR
Video system			
IP-camera		140	Technopoint
Analog cameras ($36 x 3)		108	Technopoint
Thrusters ($100 x 6)	600		CDR
Manipulator		1,000	RovBuilder
Components for tether		200	Technopoint, DZL and other
TOTAL FOR ROV	**900**	**2,845**	**3,745**
		Miscellaneous	
T-Shirts ($10x15)		150	Interface
Air Tickets ($1,800 x 6)		10,800	Biletur
Visas ($200 x 6)		1,200	Biletur
Accommodations		2,100	Booking.com
TOTAL FOR MISCELLANEOUS		**14250**	**14250**

Expenditures for Parts, Materials and Miscellaneous, accounting Donations, totaled USD17,995.

Services

Service	Donations, USD	Expenditures, USD	Company
ROV			
Water jet cutting		360	Advanced cutting technology
Turning and milling services	200		Maritime State University
Turning and milling services		50	Outsourcing
TOTAL FOR ROV	**200**	**410**	**610**
Miscellaneous			
Printing (estimated)	40		CDR
TOTAL FOR MISCELLANEOUS	**40**		**40**

Expenditures for Services, accounting Donations, totaled USD650. Thus the project's total budget is **USD18,645**. Cost of the ROV is **USD4,355**.

Time

Person	Time, hours
Angelina Borovskaia	350
Denis Mikhailov	290
Sergey Mun	480
TOTAL	**1120**

Contributors

Contributors	Amount, USD
Maritime State University	200
Center for Robotics Development	940
DNS LLC	17,505
TOTAL	**18,645**

Note: there are no re-used items, as the team is in the project for the first time.

Design Rationale

Technical requirements to the vehicle

We started designing the vehicle with identifying relevant technical requirements. So, we thoroughly studied the mission tasks and developed several use-case models. The use-case model is a document describing how and in what sequence the vehicle under development should perform mission tasks. Then we analyzed all the use-case models and selected the best one meeting three criteria: time of solving mission tasks, the ease of devices' fabrication, and the cost of component parts and production.

As a result we have defined the following main technical requirements to the vehicle.

Requirement	Target
Must have attitude stabilization	To solve tasks of exact positioning, for example, determining the object size by its image, inserting a hot stab into port, etc.
Must have a manipulator	To perform the following tasks: collecting samples, returning corroded oil pipeline section to the surface, installing a gasket, delivering the voltage meter to the place of measuring the voltage, etc
Must have a pump	To suck in a table tennis ball (algae) and discharge water into the pipeline
Must have a rotate camera in front of the vehicle	To reduce the number of cameras needed for scanning and monitoring the payload tools located on the front part of the vehicle
Must have a flange and T-bolts' fixture	To have flanges and T-bolts promptly mounted
Must have a voltage meter	To have voltage at anodes measured

Must have a turner	To have valves turned in / turned out
Gasket holder	To have gasket promptly mounted

SID (system interconnection diagram)

An important part of the work on the project is developing technical documentation, such as use-case model, technical requirements, a general electrical diagram, connection diagram, data package format, etc. These allow for coordinating the work of several people, guard against forgetting, and just simplify the work. One of such diagrams is SID.

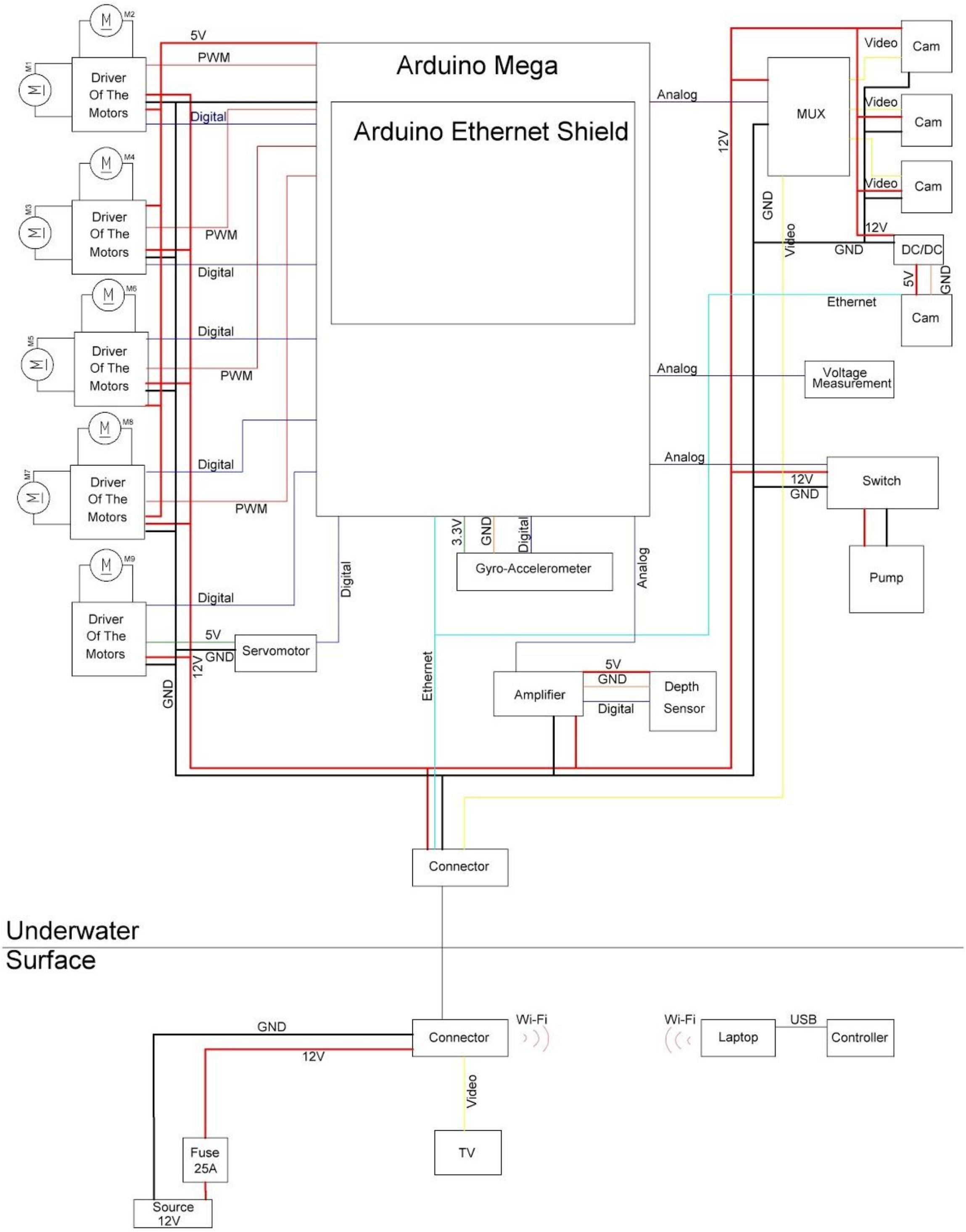


Fig.3 System Interconnection Diagram (SID) of the ROV

Vehicle system

We are in the competition for the first time, and it means we haven't had any experience in underwater vehicle designing and building. We have studied all the literature on ROV development available from our mentors, but this turned to be very complicated, not intended for school students. And we have had to learn through the process of practical work. That is why working out every single system of the vehicle has demanded from us some deep research, development and engineering analysis, studying the analogies, brainstorming, consulting our mentors. In this section we'll try to tell how and why we have developed the main systems of our vehicle.

Frame

While developing the vehicle design we have identified for ourselves three principles to rest upon in designing. Principle one: the vehicle must be functional, that is complying with technical requirements. Principle two: the vehicle must be easy to operate and maintain. Principle three: the vehicle must have an appealing appearance.

It happened, however, that the first design versions were based on the "appealing appearance" principle only. The vehicles looked beautiful, as our designers thought, but absolutely not functional. Having developed the use-case model and technical requirements we approached the design development on a less emotional but more practical basis. This let us build an ROV capable of solving all the tasks and materialize our aesthetic demands in the vehicle.

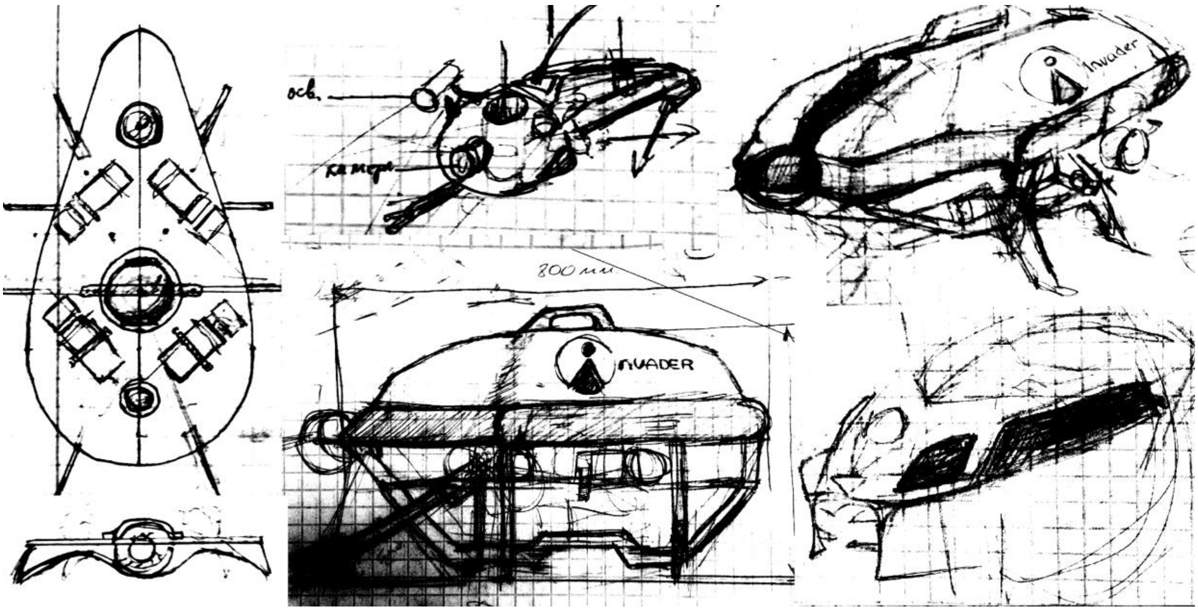


Fig.4 ROV design sketches

However, when we presented these design versions to our mentors the latter asked us the following questions: "Well, how are you going to carry the vehicle? How are you going to fix any given device? What will your actions be in case of leakage", etc. The questions made us consider the idea that we're building a robot not for an exhibition, but for real work. And we re-developed the design of our ROV once again.

Fig.5 3D-model of the frame and the vehicle as a whole

As a result we got a rather simple frame, consisting of an upper plate, a front and a rear ribs, and a keel. We chose sheet polyethylene as the material for the frame, as this material possesses increased strength, good flexibility, simplicity of processing and a good look. Furthermore, sheet polyethylene can withstand great static load. Polyethylene plates are successfully used under low ambient temperatures of down to - 40°C. And this fact is of great importance as ROV is to be operated in the Arctic.

We developed the design in SolidWorks environment. When the model was ready we prepared pattern cutting and sent it together with the material for cutting. The work was done by a third party with a water jet cutting machine.

We decided to manufacture the buoyancy plate from construction extruded polystyrene foam (density 35 kg/m3). We chose that material because of its positive buoyancy, easiness in processing and availability. The material has practically zero water absorption ability (unlike expanded polystyrene), high resistance to rotting and fungi. Extruded polystyrene service life is up to 100 years.

Building buoyancy included several stages:

1. Bonding buoyancy parts into a single plate;
2. Reinforcing the buoyancy with glass fabric cloth and epoxide resin;
3. Applying hard putty for eliminating unevenness;
4. Polishing;
5. Applying primer paint and painting.

Fig.6 Manufactured buoyancy

We painted our ROV in black and red to be distinctly seen against the white background. Final dimensions of the vehicle are: length 600mm, height 332mm, width 420mm.

Propulsion system

One of the most essential systems in any ROV is its propulsion system. Apart from that it is the most expensive one in the vehicle, which means that the thruster price can significantly affect the final cost of the ROV.

In order to find out which thrusters we would need we roughly calculated the vehicle's weight and measurement together with the payload it's going to carry. Then we calculated the horizontal (~3kgf) and vertical (~2kgf) thrust, that the thrusters would produce to move the ROV at a speed necessary for performing the missions within the time limits established by MATE. Further we had to decide whether to build thrusters of our own, or to purchase commercial-of-the-shelf ones. Having considered our capabilities we decided to buy thrusters. But we were very fortunate, as Center for Robotics Development had developed a new thruster: we got great thrusters free of charge!

Fig. 7 Thruster

The thruster dimensions are 70X40X40 mm, power feeding voltage 12V, power output 50W, forward thrust 1,6kgf, and reverse thrust 1.2kgf.

On the basis of the thruster characteristics we could decide on the number needed. We decided to install 2 vertical and 4 horizontal thrusters. Horizontal thrusters were oriented so as to provide for movements forward/backward, rightwards/leftwards and turns to the left/to the right.

First we decided to make use of Arduino LN298N.standard circuit board as motor drivers. But during debugging we understood that it wouldn't go due to insufficient for the peak-load conditions output current of 4A. Besides, it heated much, produced great noise and had big sizes. Therefore we decided to substitute it with Pololu dual vnh501 board. The output current of the board is 12A, it is 5 times smaller in size than LN298N, and this board carries out PWM operations with the frequency of up to 20 kHz, and is therefore practically silent for a human's ear.

Video system

Video system is the main sensor on our robot. The number of cameras should be enough to secure comfortable piloting the vehicle and solving various missions with the payload tools. That is why we began to decide upon the cameras' number and placement after we had decided on the payload tools' types and mounting places. As most of these tools are located "inside" ROV, we have installed 3 cameras so that they look inside. One forward, one backward, and one from above. And one more camera has been installed by us forward, but as the front camera has to perform the navigation and measuring functions, we have decided to make it a rotate one.

When choosing cameras we have been guided by the following criteria: optical sensitivity, viewing angle, sizes, absence of refraction and distortions.

In the capacity of "internal" cameras we have used AV cameras Falcon Eye FE I80C/15M, which feature big viewing angles (72 degrees), and a good matrix 1/3'' CCD Sony Effio (700 TVL, 0,001Lx).

Fig. 8 3-D model and real camera Falcon Eye FE I80C/15M

But this camera does not meet the requirements of measuring task due to intense barrel distortion of its lens and insufficient image resolution. A measuring task is understood as defining the sizes of objects as a result of image processing.

That is why we have used IP camera D-Link DCS-2103 as a front camera, as it features high data rate, high resolution, and absence of distortions. We designed and built housing and rotation gear for it. And in the end we had a rotate camera.

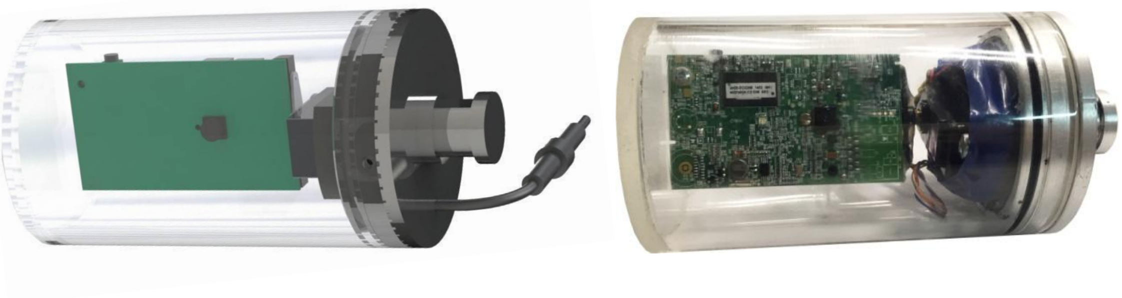


Fig. 9 3-D model and real rotate camera

Electronic unit

We have decided to accommodate all the controlling and navigational electronics in one Electronic unit. By controlling electronics we mean a microcontroller to control thrusters and payload tools, cards for securing communications. By navigational electronics we mean position sensors and circuit boards for securing their operation.

At first we planned to use Arduino Uno as a controlling board, but through the designing process we came to an understanding that we would lack pins (neither digital, nor PWM, no analog ones) and we'd have to use two Arduino Uno. Then we carefully analyzed how many pins we needed, and what requirements the controlling board would satisfy (sizes, performance capabilities) and we, after brainstorming, selected Arduino Mega, having 54 digital inputs/outputs, of which 15 support PWM, and 16 are analog inputs.

Our preference of Arduino is accounted for by the fact that of commercial boards, meeting our demands, that was the most accessible board with a great number of compatible devices, and also Arduinoes are accompanied by a great amount of teaching materials and free software for programming it.

In the capacity of networking device we have chosen Ethernet Shield. This is an expansion card installed on Arduino. It provides an opportunity to communicate with a laptop by wired network. The card is based on Wiznet W5100 microchip, which supports both TCP, and UDP protocols. To communicate between each other Ethernet Shield and Arduino use contacts #4 and from #10 to #13. All the other contacts are connected to the base board directly, that is they are actually «extensions», which is quite useful when switching on.

Pololu dual vnh501, employed for controlling thrusters, manipulator and locking, are also placed in the Electronic unit.

As for transmitting analog video to the surface only one coaxial cable is used, a device for switching between the cameras is necessary. We have developed a plate on the basis of AD8184ARZ multiplexor, that does camera switching.

For the purpose of compactability of the Electronic unit we have accommodated AD8226 amplifier on the multiplexor plate as well. We use it for amplification of the signal coming from the pressure sensor.

Sensors

Pressure sensor

As a pressure sensor we use piezoconverter D0,1T-4, which gives output voltage of 128mV at a pressure equal to 10m H2O. And this leads to the accuracy of handling by Arduino of approximately +-0.25cm. Therefore the signal needs to be amplified. By means of the amplifier we can obtain the accuracy of +-0.006m.

Accelerometer, gyroscope, thermometer

For the purposes of orientation in space and navigation we use GY-521. The sensor contains an accelerometer, gyroscope, and thermometer in a single plate. It measures acceleration and pitch angle in 3 axes: x, y, z and indicates alterations of the angle of from -180 to +180. We have chosen it thanks to reasonable price and small size. It is fed by 3.3V current and is connected to Arduino through I2C interface.

Leak sensors

Leak sensor is plate with two separated conductors. With water coming between the conductors these are short-circuited, which is detected by Arduino board.

Housing

To place cards of the Electronic unit and sensors, we developed housing for it. The housing is made of aluminum rod of 110mm in diameter and consists of a cylindrical case and a cover. Packing rings are used for sealing. We accommodated the pressure sensor there as well and twisted it in the cover from inside. Thanks to this solution we didn't have to build a separate housing for the pressure sensor.

Fig. 10 3-D model housing for electronic unit

Tether

Tether connecting ROV and surface is necessary not only for data transmission, but for providing ROV with power. When creating the tether we had a goal of making it thin and very flexible. That is why we made it a composite one of patch-cord, power cable and coaxial cable.

So in case of any cable rupture, we can replace this cable alone, avoiding replacement of the whole tether.

In order to select a power cable we have calculated resistance and drop in voltage with various cutsets. The thicker the cable, the lesser the drop in voltage, but flexibility decreases and cable weight increases. As the outcome of computations and searching the best solution we decided to choose a power cable of 3mm cutset in diameter.

To have the cables protected from mechanical damage, at the same time retaining flexibility and adding buoyancy, we decided to place the cables into a silicon tube. This have turned to be a very difficult task, and have taken several days to be achieved, a cylinder of lubricating spray, and a 9-storey building with balcony doors opened. However, despite all the hardships, including adverse weather (the ambient temperature was +5°C, the lubricant set solid and we were chilled), we did our best to obtain a flexible and light tether.

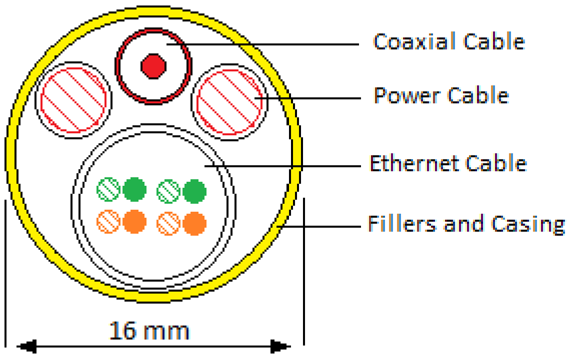


Fig. 11 Cable cross-section

Surface equipment

Surface equipment consists of a commutation unit, a joystick, two laptops, a TV monitor. The first laptop is for the pilot, all the necessary information from the vehicle and video from the front camera are displayed on it. A joystick is connected to the laptop for the sake of controlling the vehicle. The TV monitor is supplied with the videos from three "internal" cameras, which can be switched over by means of the multiplexor. The second laptop is used by the second pilot, who receives video from the front camera. The second pilot's duty is to process the images and calculate the sizes of the objects set up by MATE.

Laptops and TV monitor obtain the data from the vehicle through the commutation unit which consists of Wi-Fi router and antenna, power supply of 12V DC and a cooler, voltage meter, current meter, disconnecting breaker and connectors. Connector 2RMDT27KPN19G5V1V is used for connecting tether, two connectors RJ-45 for connecting laptops, AV connector, and power supply connector 220V AC.

Fig. 12 Commutation unit

To accommodate all these devices in a single unit we opted to use a waterproof case to carry GO PRO camera with accessories. It perfectly fits in size, secures necessary strength and is inexpensive.

Software

All the software of our vehicle can be divided into the user и on-board software.

User software

We have written the user software in C++ using free libraries SFML (window, obtaining data from the joystick, etc.) and OpenCV (obtaining images from cameras). For communications between the surface and the ROV, UDP protocol is used which perfectly fits data transmission task in a local area network, because data loss is unlikely with such quantity of hosts, while redirecting the data (as it is usual with TCP/IP protocol) would be unbeneficial. TCP/IP secures too much surplus information, which considerably slows data exchange.

The user interface comprises two programs: that of the surface (installed on the pilot's laptop) and the program for measuring object size (installed on the second pilot's laptop).

The surface window contains all the information necessary for the pilot, such as: stabilization status, leakage sensor readings, roll/pitch, and timer. During the surface's operation the contents of the packages exchanged is continuously recorded into log, which gives us an opportunity to diagnose sensors and detect deficiencies in time, also it provides us with an opportunity to accurately match controller factors. To record into the log a table format ".csv" is used, which facilitates data recording into the log. In csv format data recording is simple. A separating character for columns is ';', for lines - a line feed character. Having a tabulated log anyone even unfamiliar with programming can easily open the log file in a table editor (MS Excel, Libre Calc) and promptly plot the graph for changes in depth, roll and so on. Thanks to this solution we reduced the team's dependence on a programmer when debugging the vehicle.

The program for the second pilot lets us measure the sizes of such objects as an iceberg, corrosion, and wellhead. The program uses the following algorithm:

1. Marks in a picture the object #1, the size of which user knows. The program remembers the size of the object in pixels ($S1_{px}$)
2. Inputs to the program the real object size ($S1_{real}$).
3. The program computes the ratio of the real size to the size in pixels $k = \frac{S1_{real}}{S1_{px}}$.
4. Marks the object #2, which sizes user needs to measure. The program remembers the size of the object in pixels ($S2_{px}$).
5. The program computes the real size of the object #2 according to the formula $S2_{real} = k \cdot S2_{px}$

On-board software

On-board software operates on the on-board microcontroller. In our case it is on Arduino Mega. The environment in which we wrote the program is called Arduino IDE. It has lots of libraries to facilitate work with microcontrollers.

Initially, we wrote several modules, which we later combined into one. Some examples have been borrowed from Arduino official site. For example, data exchange UDP Send/Receive String and data processing with GY-521.

The first module that we wrote, served for network data exchange using UDP. The protocol has been partially borrowed from the official site and adjusted for our requirements. Ethernet UDP library has been used in it.

Afterwards the module for receiving and processing data from GY-521sensor has been created. Libraries for the module, namely I2CDev and MPU6050 have been downloaded from GIT-HUB and partially modified as well. A characteristic feature of GY-521 and the libraries we use is that these allow for obtaining from the sensor not raw but processed values of the pitch angles. Moreover, the program has been capable of accounting for gravitational acceleration and some other causes for errors.

The next module which we wrote was that of processing values of the data obtained from the pressure sensor. In it we read raw values from the sensor. As the analog-digital converter produced too large amplitude, we had to average out the values. The sensor itself does not show the depth; therefore the values obtained from it should be converted.

The module for motor control was written at the same time as modules of processing the data from sensors.

Then, having all the data from navigational sensors and having learnt to control the thrusters we were able to write a PD-controller. At the beginning we could write a proportional controller only, but after testing in a pool we came to understanding that for reducing over-regulation we needed a differential component as well. So we wrote it which led to an improved balancing of the vehicle, at least when in depth.

Payload tools

The most important part of our vehicle as a tool to solve specific tasks is payload. That is why we commenced the device designing from developing payload tools. We armed ourselves with the technical requirements we had developed and set to work. In solving this problem we used the brainstorming method. At first as a team, then separated into small groups of 2-3 people, and finally altogether again. And when we gave birth to some, in our view strokes of genius, we invited our mentors for discussing our ideas. But they didn't like a single idea. That was terrible. And we had to go through all the stages anew, this time paying regard to the feedback from the mentors. After several stepwise approximations we eventually arrived at acceptable solutions.

Manipulator

We decided to buy a manipulator produced by ROVBuilder Company, to perform the majority of manipulation tasks with it: collecting samples, returning corroded oil pipeline section to the surface, installing a gasket, delivering the voltage meter to the place of measuring the voltage, etc. We opted this manipulator because of its relatively low cost, high production quality, and reliability. Furthermore, this manipulator seems be the only underwater grabber available at the Russian market. We refrained from an attempt to build a manipulator of our own due to lack of experience.

Fig. 13 ROVBuilder manipulator

Rotate camera

In the front of the vehicle we needed a camera to watch the manipulator grabbing and look forward for the purposes of machine vision. We decided to combine these cameras into one and developed a rotate camera capable of rotating 180 degrees in vertical plane.

Valve Turner

Within two missions the vehicle is to turn in/turn out valves. We first planned to install 2 pintles on the vehicle and having ridden the valve to turn the whole robot. But then, after computing the time it would take considering cable unwinding, we decided to manufacture a separate device to promptly turn in/turn out the valves. We named the device the Valve Turner. It consists of a motor, waterproof casing, and a head piece made of a single-part aluminum plate.

Fig. 14 3D-model and real Valve Turner

Pump

In mission three we have to move water through a pipeline. To have this completed we decided to use an oil transfer pump because of its small size: diameter of 52mm, length of 140mm, voltage of 12V and power output enough for completing the task. Once we've got a pump we decided to use it for collecting samples of algae (a table tennis ball). For the purpose we placed a hopper, made by us, in the upper part of the vehicle. The hopper is connected to the suction side of the pump and has special thin plastic leaves for retaining the ball.

Fig. 15 Pump with head pieces

Flanges holder

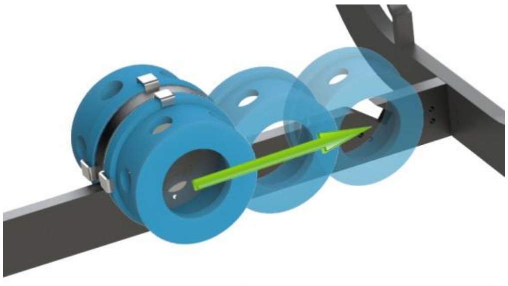


Fig. 16 Principle of Flanges holder

We set ourselves a task of installing the flanges, T-bolts, and a gasket so as to eliminate the necessity for floating up to the surface for each of them separately. For the purpose we developed 3 different types of holders: flanges, T-bolts and gasket holders. The holders are to keep a part and let it go when necessary. Apart from that in order not to complicate the vehicle in terms of electronics and software we decided to make them purely mechanical.

Flanges holder holds a flange from three sides in such a manner that when we install it on the tube, it can undock from the vehicle on the fourth side.

T-bolts holder

T-bolts holder is very simple in design. It holds the T-bolt by means of two elastic steel plates. And once we insert it into the flange opening we can, having moved aside, undock it.

Fig. 17 3D-model of T-bolts holder

Gasket holder

Fig. 18 Gasket holder

It consists of 4 elastic steel plates to hold the gasket. However, when we will install it the wellhead walls would draw the plates apart and the gasket would fall inside. This holder is mounted on the manipulator.

Voltage meter

To have the voltage on anodes measured we developed in SolidWorks and built from sheet polyethylene a special Voltage meter. It is so designed as to «brace» all the anodes simultaneously on the leg of an oil platform.

Fig. 19 3D-model of Voltage meter

Safety

Our philosophy as far as safety is concerned can be described very simply:"Safety First". For this reason we had different 4 safety instructions:

- For operating electromechanical equipment in a workshop;
- For work in an electric-installation workshop;
- For work in a pool;
- For operating the vehicle.

We would give Safety instruction for operating the vehicle as an example:

Safety Checklist

Pre-start check	√
Check fuse	
Check all connectors and cables	
Check mounts	
Check current and voltage	
Check leak sensors	
Check data from ROV sensors	
Check cameras	
Check thrusters and manipulator	
Monitoring during operation	
Monitoring leak sensor	
Monitoring data from ROV sensors	
Monitoring Current and Voltage	
After-finish check	√
Check mechanical damage	
Check all connectors and cables	
Check the structural integrity of thrusters and propellers	

To secure the safety the ROV possesses some safety features: a leakage sensor in Electronics unit, sound alarm for leakage, a fuse on the cable, the frame itself is designed so as to minimize the risk of mechanical damage in its “internal contents”. Also the ROV has safety features to provide for the safety of the personnel operating and maintaining the robot: cautionary labels, hoods on thrusters, special handles for carrying the vehicle.

Challenges

Technical

While writing an interface program the design presented a great challenge, as the programmers used to be engaged in competitive programming and they encountered the task of developing a user interface for the first time. Questions arose as to correct location of the elements within the window, so that the pilot would see all the necessary values without distracting his attention from the vehicle control. To have the challenge met we turned to interfaces used in previous competitions and to our mentors. We had to not only learn how to ergonomically locate buttons and captions on the screen, but to dramatically change the attitude to software style, think of a user and effectiveness of using our program by other people. In competitive programming we were required to solve the task by finding the fastest most reliable algorithm. No users, algorithms alone.

We made a few scratches on paper, having learnt the experience from our mentors, discussed the convenience of use with every team member, everyone spoke out and eventually determined the final design.

Non-technical

We have never worked as a team before, apart from that we go to three different schools and reside in different parts of the city. So, the greatest challenge for us was poor coordination of our work. There were cases when several people were doing the same work, while no one was engaged in solving the major task for that stage. Also, there was a hap when electronics engineers decided to remake a plate, but had not updated mechanical designers on thier decision, and the latter continued to design the frame according to the previous size of the plate. The result was that the plate couldn’t be accommodated into the frame, while we lost time for making alterations.

We didn’t find solution to the problem at once, and moreover we even failed to see it until our mentors drew our attention to it. And after that the solution came easily, though it was difficult to put it into action. We made up timelines of work for the team and every separate group. We started to hold general meetings two times a week. We established a shared folder with current drawings, diagrams and protocols. This not only let us solves our problems, but also made us more disciplined and efficient.

Troubleshooting Techniques

Each group employed its own troubleshooting technique.

Programmers followed the next algorithm:

1. Make sure the problem is that of software and not of hardware.
2. Trace the failure. This is usually done with small tests for each module.
3. Understand the essence of the problem.
4. Find and employ a new solution. For the purpose various methods are used, including brainstorming and extreme programming.
5. Test the module and the program as a whole. Should the failure remain, proceed to item 1.

Electronics engineers had an approach of their own, but the main stages are similar to the stages of the program debugging:

1. Make sure the problem is that of electronics, and not of software nor design.
2. Trace the failure. A tester is the first thing to be used for the purpose.
3. Understand the scope of touchup: if it is enough, for example, just to re-solder component parts, or the plate should be re-manufactured as a whole.
4. Search for the best solution and its deployment.
5. Do the testing.

An algorithm for troubleshooting pertaining to the design and payload is similar to the above algorithms. The main thing that is different is the method of tracing the failure.

Overall testing of the vehicle is carried out according to the following algorithm:

1. Check power supply. When the power is supplied lights on Arduino should show up;
2. Check communications. When wired to the vehicle, Pilot console receives data from it;
3. Check leak sensor. Leak sensor to be closed in turn and received signal is checked.
4. Check thruster system and manipulator. Supply the power to each device independently and monitor its operation correctness.
5. Check video system. Video captured from each camera to be checked independently, switch between cameras' streams to be checked.
6. Check navigation sensors. The vehicle is moved in 3D by a command, sensors to reflect the changes in position.

Future improvement

Computer Software

The weak point of our vehicle, to be more exact, of its surface equipment, is, for the time being, the joystick with which we control the vehicle. Should it break finding a similar model can take lots of time, which will result in an undesirable idle period of the equipment. Currently, if we substitute the joystick with another model, we'd have to re-write the program.

In future the surface software should be supplemented with a window for the joystick configuration, with replacing some of the buttons with keyboard keys, in case of their lack. This will release from heavy dependence on the joystick, and substitute it in case of failure.

Manipulator

A manipulator now is the most expensive commercial device employed in our robot, this being said, it isn't very suitable for completing a number of tasks that we solve. Therefore the work is done at a slower rate compared to the desired situation of using a manipulator of our own specifically designed for these tasks. We plan next year to develop our own inexpensive manipulator featuring various head pieces for completing a range of tasks.

Lessons Learned

Technical

Programmers

There are only 2 people in the programmers' group. At first each of us was specializing in his part of work only. Daniil was busy with programming Arduino, and Nikolai - with programming User software. This let us perfectly concentrate on our work, but when any of us fell ill or was away, the whole work could stop. And everyone blamed the programmers on the standing time. Such cases were rare but these let us learn a useful lesson: with few people in the team, the team members must capable of changing off. This lesson made Daniil learn how to program in QT programming environment, and Nikolai - how to program in Arduino IDE.

Mechanical designers

We improved our skills in operating such electrical tools as Angular grinding machine, Dremel, Drilling machine. We learnt how to work with SolidWorks. We got better understanding of all the technical details of engineering, for instance how to correctly execute drawings, comply with the measurement convention system.

Interpersonal

Teamwork taught us something greater then working skills and communication skills. We understood that a team is a mechanism like a robot, but only a social one, where in place of component parts and program code there are people and their team spirit. To make a component part fit other ones; it should be machined, and smoothed. In a similar manner, people should get used to each other, and find common language. And in order to make the team come alive, like a robot comes alive upon a weaving process, a team should obtain a team spirit, which is a task to work upon by every team member.

Reflections

Anton Iakovivskii. This year our team has been preparing for participating in MATE ROV Competition. During the preparation period I learnt how to do many things and obtained a lot of knowledge. I got familiarized with working in SolidWorks, with designing 3D models. I learnt how to operate various equipment such as dremel tool, drilling machine, sawing machine, etc. I learnt how to operate them and use for making props and assembling the ROV. I enjoy working in a workshop. I hope that our team will pull ahead. It's a challenging task demanding much time and effort. Our mentors are always eager to help us. I like working in a team, to learn from the senior colleagues, feel myself competent, if I can render assistance to someone else.

Lev Davydenko. Preparation to these competitions turned out to be one of the greatest challenges to myself lately. Maybe the most difficult task within the span of my life. We spent lots of time and effort. We failed at many undertakings, as the things we did during preparation were absolutely new to me. For example, soldering, it was very difficult for me, because I couldn't do it before, and to tell the truth I am not good at it even now, I have to re-solder many times, which is very time and effort consuming, but I'm doing my best. Sometimes more experienced team members help me, and I'm thankful to them for this. I'm sure that working in such a team of excellence would help me grow into an engineer of excellence.

Team. Self-criticism underlies our personalistic, professional, and team reflections. At each meeting we try to identify not only positive dynamism in our joint work, but negative trends as well. We call this methodology: «3 pluses and 3 minuses». Each teammember is to name 3 improvements and 3 areas leaving the space for improvements. Thus we can avoid «star fever» and be continuously concerned about the development of ourselves and of the team.

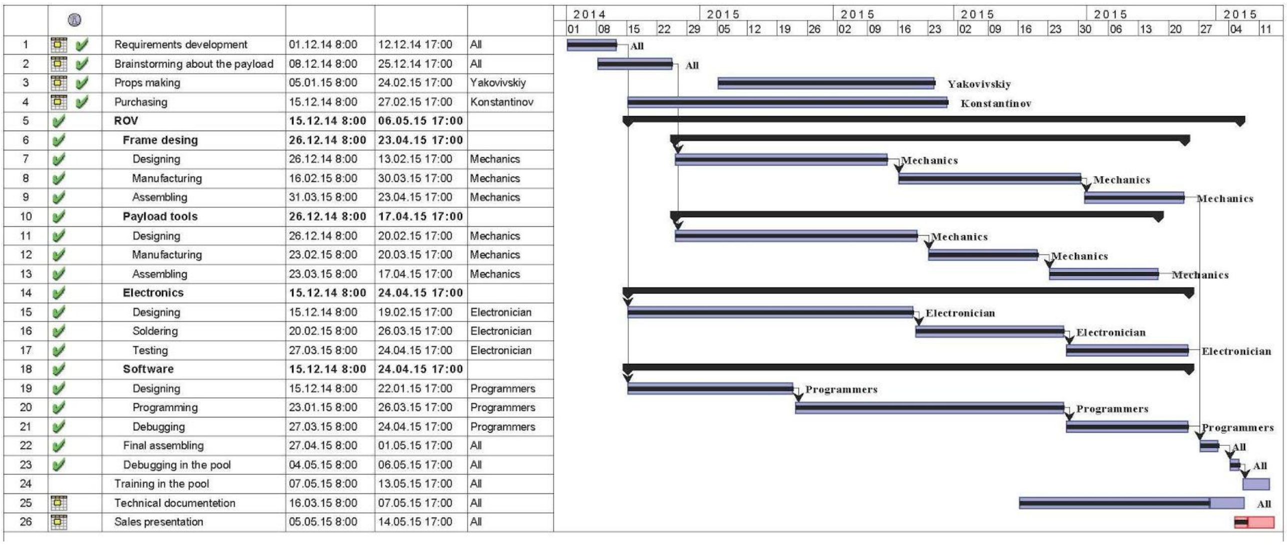
Teamwork and Project Management

We are eight in our team. To have the work done we've divided into groups: three mechanical designers, three electronics engineers, and two programmers. Everyone completes his tasks: mechanical designers make props, design the vehicle itself in Solid Works and assemble it, electronics engineers work on electronics to be located in the vehicle and with the help of which it's going to be controlled, and programmers work on software. Each group is headed by a chief to coordinate the group's work. Each group is supervised by its mentor. It's mentors' duty to teach us the basics of designing, work in CAD, work with tools and machines, acquaint us with peculiarities of underwater technologies, share experience in the field of project management, in getting ready for competitions, trainings in the pool. Nevertheless, all the work in designing, building, and debugging the vehicle was done by us on our own.

Apart from being divided into groups, every team member has a project role. Anton Konstantinov is CEO and CFO all in one. It is his duty to monitor the general course of work, to purchase the components and to do bookkeeping. Marat Rasulov, CTO, is responsible for the work with complicated machines and tools. Nikolai Dolzhenko, COO and CIO, is in charge of keeping the team members updated and of computing devices' operational integrity. Matvei

Kostylev, CMO, is responsible for the style of the vehicle, technical report and poster. Lev Davydenko is in charge of safety check and of developing electrical diagrams. Anton Iakovivskii is responsible for timely manufacturing of props. Iaroslav Proshin is in charge of devices' waterproofness. Daniil Proshin is engaged in editing the technical documentation and meeting minutes keeping.

To have our work better coordinated our mentors have advised us to make our timelines of work in the form of Gantt chart. For the purpose we used ProjectLibre freeware.



1	Requirements development	01.12.14 8:00	12.12.14 17:00	All
2	Brainstorming about the payload	08.12.14 8:00	25.12.14 17:00	All
3	Props making	05.01.15 8:00	24.02.15 17:00	Yakovivskiy
4	Purchasing	15.12.14 8:00	27.02.15 17:00	Konstantinov
5	**ROV**	**15.12.14 8:00**	**06.05.15 17:00**	
6	**Frame desing**	**26.12.14 8:00**	**23.04.15 17:00**	
7	Designing	26.12.14 8:00	13.02.15 17:00	Mechanics
8	Manufacturing	16.02.15 8:00	30.03.15 17:00	Mechanics
9	Assembling	31.03.15 8:00	23.04.15 17:00	Mechanics
10	**Payload tools**	**26.12.14 8:00**	**17.04.15 17:00**	
11	Designing	26.12.14 8:00	20.02.15 17:00	Mechanics
12	Manufacturing	23.02.15 8:00	20.03.15 17:00	Mechanics
13	Assembling	23.03.15 8:00	17.04.15 17:00	Mechanics
14	**Electronics**	**15.12.14 8:00**	**24.04.15 17:00**	
15	Designing	15.12.14 8:00	19.02.15 17:00	Electronician
16	Soldering	20.02.15 8:00	26.03.15 17:00	Electronician
17	Testing	27.03.15 8:00	24.04.15 17:00	Electronician
18	**Software**	**15.12.14 8:00**	**24.04.15 17:00**	
19	Designing	15.12.14 8:00	22.01.15 17:00	Programmers
20	Programming	23.01.15 8:00	26.03.15 17:00	Programmers
21	Debugging	27.03.15 8:00	24.04.15 17:00	Programmers
22	Final assembling	27.04.15 8:00	01.05.15 17:00	All
23	Debugging in the pool	04.05.15 8:00	06.05.15 17:00	All
24	Training in the pool	07.05.15 8:00	13.05.15 17:00	All
25	Technical documentetion	16.03.15 8:00	07.05.15 17:00	All
26	Sales presentation	05.05.15 8:00	14.05.15 17:00	All

Fig. 20 Gantt chart

Fig. 21 Lev and Marat are making buoyancy

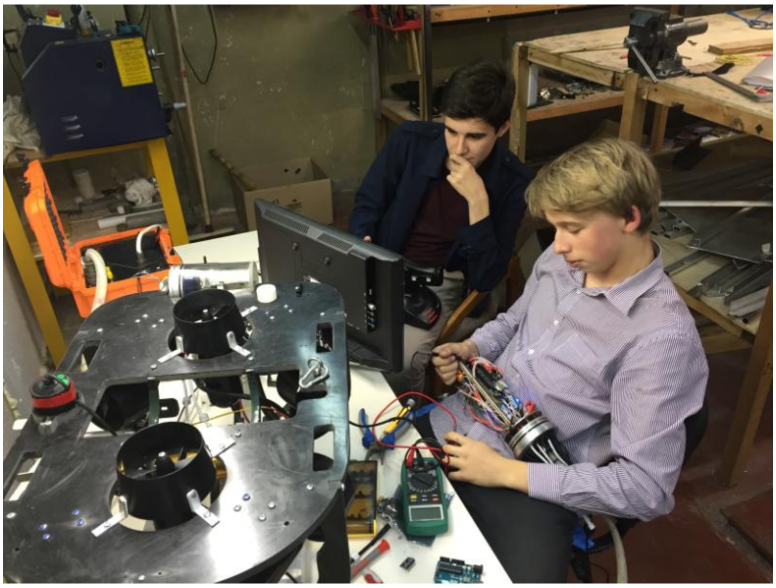


Fig. 22 Anton and Nikolai are debugging ROV

Acknowledgements

We would like to extend our heartfelt gratitude to MATE Center for the opportunity to participate in the competition, and for the interest that has united out team in pursuing the shared goal, as well as for any support rendered.

We thank the Center for Robotics Development for financial, administrative, technical, and moral support.

We are clearly thankful to all our mentors who spared no effort and time, with great patience and care helped us prepare for the competitions.

We would also show appreciation of DNS Computer Center LLC and personally its CEO Dmitry Alekseev for financial backup and ROVBuilder LLC for the discount on the manipulator granted.

Besides, we would like to express grateful acknowledgement to Professor Alexei Strelkov for his help with translation.

References

1. "Project Management Institute PMI - PMBOK (4th Edition)". 2008, ISBN 9781933890517.
2. "Underwater Robotics: Science, Design & Fabrication" Dr. Steven W. Moore. 2010, ISBN 978-0984173709.
3. "The C Programming Language" Kernighan and Ritchie, ISBN-0-13-110362-8.
4. "The Art Of Electronics: 2nd Edition" Paul Horowitz and Winfield Hill, ISBN-13: 978-0521689175.
5. "Introduction to Arduino" Alan G. Smith, ISBN-13: 978-1463698348
6. "Arduino CookBook" Michael Margolis and O'Reilly Media, ISBN 13: 978-1-4493-1387-6
7. "Sealing and compaction equipment" Golubev and Kondakov, ISBN: 5-217-02410-0
8. http://www.marinetech.org/rov-competition-2/

Appendix 1 Flowcharts

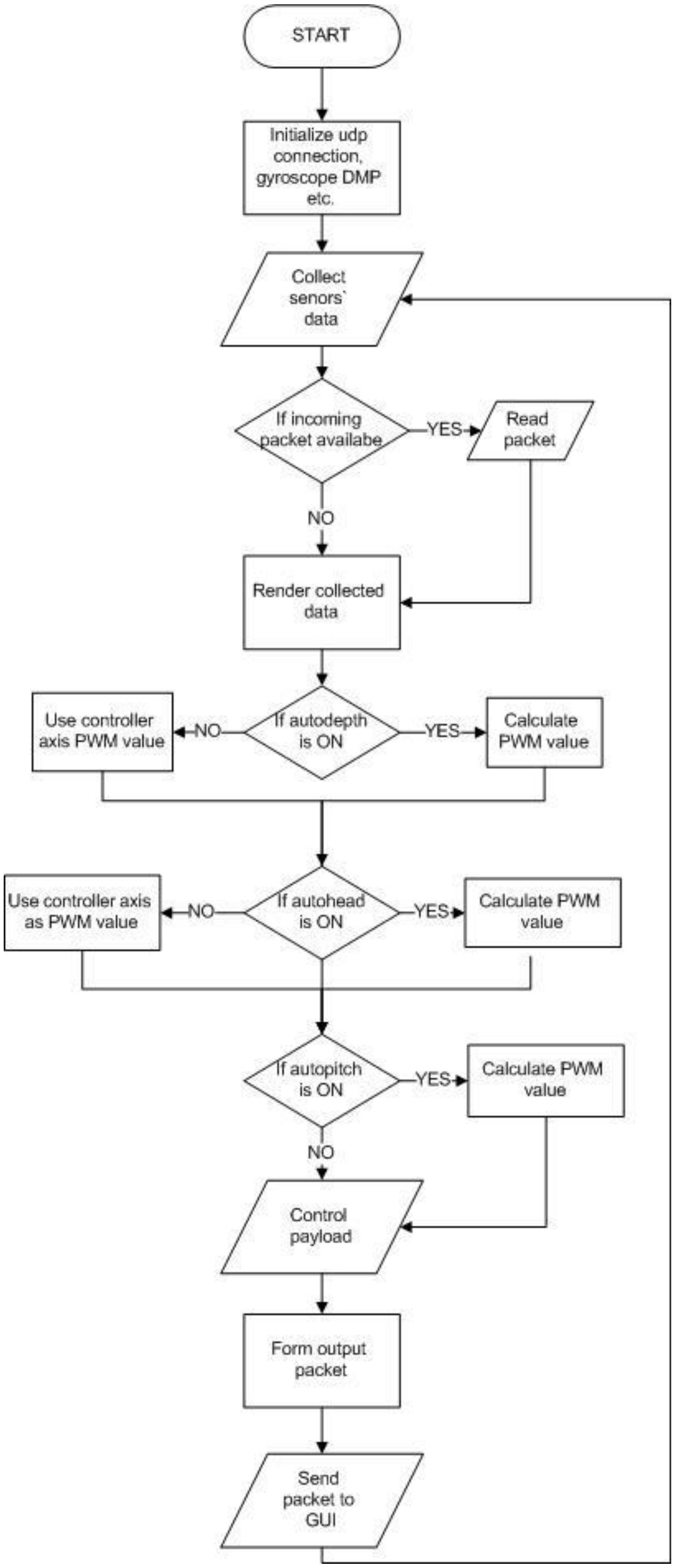


Fig. 23 Flowchart on-board software

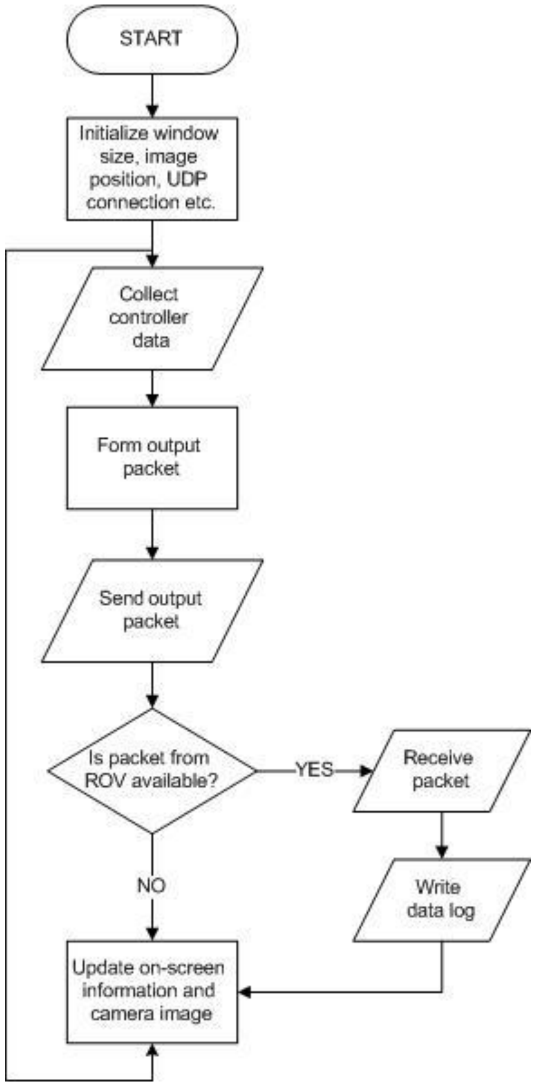


Fig. 24 Flowchart User software