Development of a Low-Cost Maritime Educational Robotics Platform

Reinhard Grabler, Markus Klein, Thomas Fellner, and Gottfried Koppensteiner

***Abstract*—The underwater environment is challenging due to the available three-dimensional space and the risk of water damaging electronic and mechanical parts. Using 3D printing technology, a housing as well as the functional parts for an underwater robotics kit can be produced low-priced at home or in schools and other facilities. Other mechanical parts used in the design are standardized and available in common home improvement stores. Conveniently the robot is designed modular, facilitating easy assembly and the possibility to test the functionalities of each module outside the housing. The used Hedgehog robotics controller is focusing on low costs in the hardware, and is extended by high usability in the software for this underwater robotics kit.**

***Index Terms*—Underwater robotics, educational robotics, low-cost, open source, open hardware, STEM, rapid prototyping, 3D printing, robotics controller.**

I. Introduction

Since about 71 percent of the Earth's surface is covered by oceans it is the largest habitat for animals [1] and 95 percent are still unexplored [2]. The world's oceans are the habitat for about one million species, with between one-third and two-thirds of it still unknown [3], [4]. In addition to that, 70 percent of Earth's oxygen is produced by ocean phytoplankton [5]. Conserving and exploring the marine environment as well as protecting marine life are objects in the interest of humanity for the persistence of our planet.

It has long been recognized that the employment of underwater robots has important practical significance, which includes their usage in ocean and marine environmental research, pipe survey, submarine and under-ice exploration, mine reconnaissance, dam inspection and so on. Over the last decades, with the fast development of underwater robotics, there have been many remarkable achievements and it has become a significant tool for the development and utilization of sea resources [6], [7]. Nonetheless, high maintenance, manufacturing, development and research costs as well as systems complexity associated with underwater robots have led to a slower adoption rate and have prevented their wider application.

Considering a shortage of student interest in the STEM (Science, Technology, Engineering and Math) domains underwater robotic kits can be especially applied in education as an innovative tool for teaching basic science and engineering concepts [8]. On the one hand, robotics integrates skills needed for designing and constructing machines, computers, software, communication systems, but also offers students the flexibility to develop interdisciplinary projects and to discover exciting topics such as movement, navigation, coordination, grasping, cognition and many others [9]. On the other hand, the usage of these kits would enable students to explore marine science and engineering principles relevant in water. Moreover, this kind of robotics kit can be used to inform future generations about the importance of the ecosystem and inspire them to do research in this field. The Submarine Hydrodynamic Autonomous Robotics Kit (SHARK) aims at providing the platform for enthusing not only as many young tech-savvy students but also as many technology-freshmen as possible into the STEM fields and also into the subject of underwater robotics. It uses the Hedgehog Educational Robotics Controller in order to allow simple and intuitive programming using a web based integrated development environment (IDE) with real-time collaboration support. To make it even suitable for elementary and middle school students, a visual programming environment is embedded as well. For more experienced users, programming is possible via the Python programming language.

II. State of the Art

Robotics projects are transdisciplinary, improve the students' skills in programming, mechanical engineering, electronics and teamwork, and as a result engage the learning of STEM concepts [10]. The positive effects of robotics to the interdisciplinary learning experience of students have been shown in several studies. However, educational robotics is still in its infancy and a lot more experience and expertise has to be shared [11], [12]. The Botball®Educational Robotics Program organized by the KISS Institute For Practical Robotics (KIPR) is one of the many successful robotics initiatives. At the Global Conference on Educational Robotics (GCER) conference students can participate in a non-destructive robotics competition. During the conference students are also able to listen to talks by scientist giving an insight into their robotics research. Unfortunately, the Botball® robotics kit is very expensive and costs about USD 2,500.

A. Underwater Robotics

The underwater environment challenges even the most experienced robotics engineers since there is three-dimensional space available and the robot is under constant risk of being damaged by intruding water. Also, the

Manuscript received January 24, 2017; revised June 5, 2017.

Reinhard Grabler and Gottfried Koppensteiner are with the Practical Robotics Institute Austria, Wexstraße 19-23, 1200 Vienna, Austria (e-mail: grabler@pria.at, koppensteiner@pria.at).

Markus Klein and Thomas Fellner are with the TGM, Vienna Institute of Technology, Wexstraße 19-23, 1200 Vienna, Austria (e-mail: m@mklein.co.at, fellnertho@gmail.com).

doi: 10.18178/ijmmm.2018.6.3.377

educational effect of underwater robotics should be pointed out. Besides gaining engineering skills due to the interdisciplinary characteristics, the interest of young people for the marine environment should be raised. GUPPIE [13] is an educational underwater robotics platform which aims to teach control concepts from middle school to undergraduate level using robotics as an interdisciplinary tool [10]. The partially 3D printed robot with a length of 60 cm and a weight of 3.5 kg can reach up to 4 m depth and has endurance of 2 h [13]. The overall cost of the GUPPIE prototype is USD 1000 [14].

Another platform is WaterBotics® which is based on LEGO® MINDSTORMS® NXT kits. It can be put under 1 meter of water for up to 30 minutes [15]. Unfortunately, the used LEGO® Power Functions Motors are not waterproofed and since they are in direct contact with water they will corrode over time and stop working.

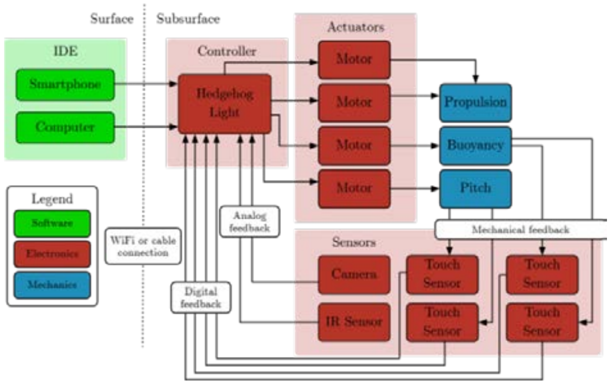


Fig. 1. Overall concept.

B. *Educational Robotics Controllers*

1) Wallaby controller: Wallaby is the robotics controller of the Botball® Educational Robotics Program based on an ARM processor running a Linux operating system. Moreover, all of the controller's software is available as open source software licensed under the GNU General Public License 3.0. Software for the controller can be written in the C programming language using a web based IDE. Unfortunately, it does not provide a version control system leading to poor usability regarding the development of complex programs.

2) LEGO® mindstorms® NXT: The NXT controller is programmed using the MINDSTORMS® software [10]-[16]. Using a graphical programming language instead of a text based language like C, helps students to focus more on the programming concepts rather than the syntax of a language, and reduces the entry level into the field of programming [16].

C. *Summary of Existing Approaches*

Despite the rise of 3D printing in the past years the underwater robotics projects at the state of the art do not consider using this technology as a main design tool for the waterproof housing or other functional parts. Moreover, existing approaches often lack either the simplicity or the extendability of the robot controller's software platform.

Hence, the goals of the SHARK are defined as follows:

- The solution should be low-cost, in order to introduce it to a widespread audience. The housing should be printable on conventional 3D printers to enable printing at home, in labs or in other facilities and the models therefore should be available as open-hardware. The flexibility of 3D printing allows to easily adapt changes of the hardware to a new housing.
- The non-3D printed parts should be standardized and available worldwide. If possible, most parts should be 3D-printable.
- The robotics controller should be exchangeable and the robot should therefore be usable in all of K-12 education. • To enable further development, all parts of the robot should be available as open hardware and the used software should be open source.
- To facilitate agile movements in small competition tanks, it should be under 40 cm in length. The robot should be designed to be used in educational robotics competitions. By using components of the Botball® Educational Program, it should be suitable for participants of the program.
- The robot should be autonomous only to promote the programming and debugging skills of the participants of the competition.
- As the system should be both attractive for technology-freshmen and adaptable for more experienced users, its programming must be simple but not restrictive at the same time.

III. Concept

The robot's controller, actuators, sensors and mechanical parts are located inside a 3D printed waterproof housing. That way those components do not need any treatment such as coating with epoxy for protecting them from water, and thus can be replaced easily. Using the 3D printing technology, the design of the robot is also very flexible and can be adapted for different components. The user programs the robot in an IDE. For the Hedgehog Light controller, the IDE is running in the browser of a computer or a smartphone. For the controller there is a library of commands which allows easy programming of the robot. The commands involve the powering of motors and reading out analog and digital sensor values. The code is transmitted to the controller via a wireless or a USB cable connection, where it is compiled to executable binaries. The transmission has to be performed on the water's surface, since 2.4 GHz wireless connections will not work throughout the water [17]. Since the controller is inside the robot's housing the program has to be started remotely from the OS. When diving, the robot will lose the connection, hence it can just be used autonomous. There is no possibility to see a debugging screen or recognize malfunction from the moment the connection is lost. For the user the controller is a blackbox, meaning the user does not have to understand its internal workings. The controller comes with a battery for power supply and the necessary high-level and low-level OS. Also, the electronic control of sensors and actuators is an already implemented solution. The used electronic components must be compatible with the used Hedgehog Light robotics controller. Also, to make it easier and cheaper to use the product for students and schools which already own a Botball® Educational Robotics kit, only parts included in the kit are used. In addition, those components are conveniently also compatible with the Hedgehog Light controller. The robot has two TowerPro SG-5010 Double Ball Bearing Servo motors, modified for continuous rotation, for propulsion and steering. If those motors are powered differently it can move in a curved path with various curvature radius. Furthermore, there is one motor that is used to control the buoyancy, and another one in order to control the robot's pitch. In both cases the rotation of the motor is transformed into an axial force by using a threaded rod. Since the starting position of the moving parts in the robot are not known, there is the need for touch sensors at both end positions of the buoyancy, respectively the pitch module. When those digital sensors are pressed, the motor of the particular module has to stop, to avoid internal mechanical damage. The digital value is sent to the controller and can be read out in the code. The camera and the IR sensor provide analog feedback. The SHARP IR sensor works as a combination of a Position Sensitive Detector (PSD) and an Infrared Emitting Diode (IRED). The method used is called single point optical triangulation and results in a value in between a range qualifying the reflection of light. Considering the absorption coefficient of water at 20° C which is $\alpha = 0.07\ cm^{-1}$, at a distance of 10 cm, the absorption is approximately 50% [18], [19]. Despite this high absorption value the sensor was tested in underwater experiments successfully and can be used in small ranges for the application. The Ubisoft Motion Tracking USB camera, originally designed for the Nintendo Wii platform, has an 85° angle of view, which is why objects can be close to the camera and still be recognized. The feedback of the camera is video information including the color and brightness for each pixel. The analog values can be applied in the code by using commands from the controller's library.

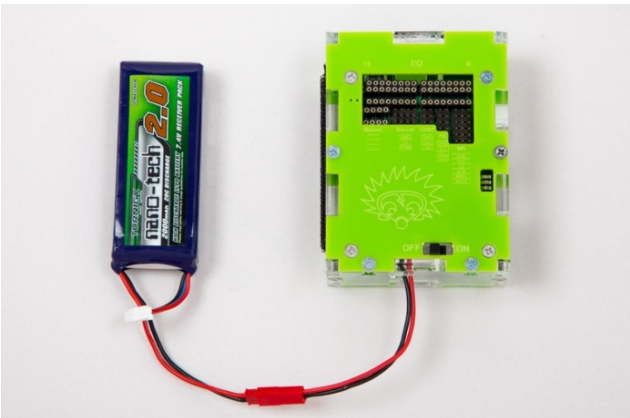


Fig. 2. Hedgehog light robotic controller.

A. Hedgehog Light Robotics Controller

The Hedgehog Light controller, an educational robotics controller developed by PRIA, consists of a Raspberry Pi 3, a PCB shield, a step-down converter, a rechargeable battery, a charging unit and a case specially designed to hold the components. In the SHARK the controller is used without its case to save space. The PCB-shield, the main hardware controller, contains a STM32 F4 micro controller with ARM Cortex M4RISC-Core as main processing unit. It handles time critical control functions, like the control of the 4 motors, 4 servos and readout of up to 16 sensors (8 digital and 8 analog) as well as the communication with the Raspberry Pi and voltage monitoring to prohibit depth discharge of the Lithium Polymer battery. Via the USB 2.0 port on the Raspberry Pi, a camera can be connected. The controller is programmed directly on a smartphone or on a PC, then the program is sent via a WiFi connection to the Raspberry Pi 3 where it is executed on the software controller. The Hedgehog Light hardware costs are approximately EUR 200.00 but it is also available as open hardware [9], [20], [21].

B. Software Platform

The software platform of the Hedgehog Light controller, and thus of the SHARK, can be divided into two main components:

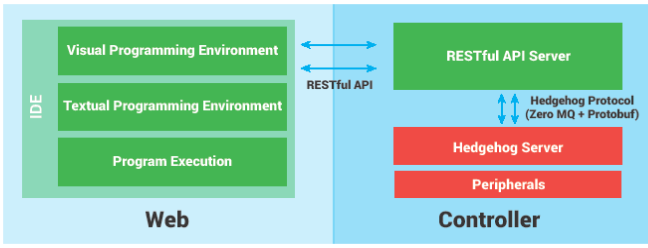


Fig. 3. Development environment concept.

- User libraries for accessing the robot's hardware
- Development tools for writing, deploying and executing programs on the controller

Usability of the controller's development tools is a very important factor to the overall usability. Especially in workshop situations with students, the development tools need to be both simple to setup and intuitive to use. However, the system should not restrict more experienced users in their opportunities.

The abilities of web applications link with techniques like HTML, JavaScript and a frontend application framework as well as the performance of browsers and JavaScript engines have steadily increased, making the web an attractive competitor to old fashioned desktop application. In fact, browser applications offer numerous advantages:

1) Browser application are naturally platform independent
2) Web apps run on both, desktop computers and mobile devices, like tablets
3) No installation is required on client side
4) All data is being stored on the robot controller, hence, programs can be accessed from multiple devices

Consequently, Hedgehog's integrated development environment (IDE) is a web application consisting of a web server running on the controller and an Angular [22] application on the user's web browser. For the communication between server and client, a RESTful HTTP service [23] is used.

a) User libraries: Hedgehog implements a protocol for interacting with the robot's peripherals (motors, sensors, servos, etc.) based on protocol buffers and ZeroMQ [24] and a Python library as reference implementation. Therefore, user programs written in Python are already supported. The authors think that Python as a dynamic programming language with a simple syntax is a good choice for beginners. In order to be attractive for an even broader spectrum of students, and to particularly address elementary and middle schools, a visual programming environment is embedded into the software platform of Hedgehog as well. Students are also able to quickly test a motor or read sensor feedback using the web interface. Due to the fact that ZeroMQ is not supported on web browsers, parts of the Hedgehog protocol are exposed through the REST interface by the controller's web server.

b) Development environment: User programs are stored on the controller and managed as *git* repositories. The web IDE itself only supports a simplified view to the underlying version control by showing the program's version history as a single version tree using the master branch HEAD as root. A program version translates to a commit on the master branch. The phrase commit and version will be used synonymously. Nevertheless, advanced users can directly interact with the *git* server of the controller and therefore use all features of *git*. In order to help students collaborating on a single program, which is especially useful during workshops or competitions, the IDE also allows concurrent editing of a single file using an operational transformation [25] algorithm.

IV. Implementation

A. Modular Design

The main reason to design the robot modular, is the assembly. Each module controls a specific task of the robot, and can be assembled and tested without being installed in the waterproof housing. The housing consists of two halves, which are attached together using a flange (see Fig. 4). The robot is structured into the *buoyancy module* to control diving, the *pitch module* to control the robot's pitch and the *prop modules* for the propulsion. The modules are simply mounted into the housing and the cables are connected with the controller, before the housing is closed. Since the controller is not accessible when the housing is closed, it has to be turned on before the robot is closed and sealed. The programs can be started via wireless connection, but only on the surface, the connection will be lost when the robot is underwater. The robot can be used autonomously only, in order to promote programming skills among students. Additionally to the option of using sensors to interact with the environment, solutions for path-planning [26] can be applied for better results.

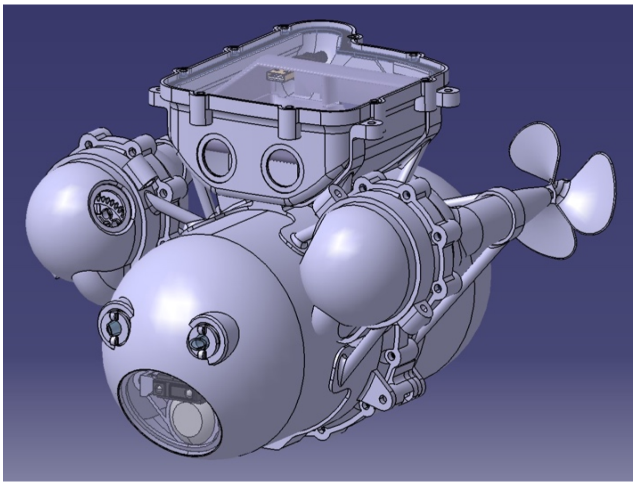


Fig. 4. 3D model of the SHARK.

a) Buoyancy module: To allow diving, the specific density of the robot has to be increased. In the buoyancy module, there are two syringes of 50 to 60 ml capacity each, which suck in water from the environment. A motor can move the pistons of the two syringes simultaneously. The pistons are connected to wingnuts which cannot rotate, since they are mechanically blocked by a 3D printed track which is lubricated to decrease friction at the connection point. That way the wingnuts are guided along a threaded rod, which is rotated by a motor. If the robot should hold a specific depth, the water must be sucked in and expelled continuously.

b) Pitch module: To control the pitch of the robot, the center of gravity can be changed. Therefore, a weight can be moved in axial direction on a threaded rod. The weight shell is designed to hold 1 kg standardized lead used by divers. This module is necessary, since the syringes of the buoyancy module are located in the front area of the robot. When sucking in water, the weight in the front changes and the robot is not aligned horizontally anymore. Also the robot can dive and hold a depth when it is tilted forward and the propellers are powered without making use of the buoyancy module. The module also holds the camera as well as the IR sensor.

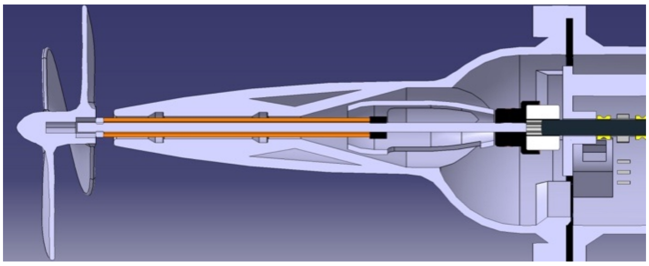


Fig. 5. Sterntube for propulsion.

c) Prop module: The prop module generates the propulsion of the robot, and also enables steering. Each motor is connected to a 40 tooth LEGO[R] gear, using a standard servo plate. Over the 8 tooth gear the rotation is transmitted to the shaft coupling. A sterntube, consisting of a shaft, a pipe and a stopper, is used to conduct the shaft rotation outside the housing (see Fig. 5). The space between the pipe and the shaft is filled with grease to prevent water from coming inside and reduce friction. The right-handed propeller is screwed onto the shaft, which has a thread of type M3 at its ending. For the design of the propeller, already constructed propellers with known efficiency were taken as references [27], [28]. With the optimized size and pitch angle calculated [19], the propellers were also 3D printed.

B. 3D Printing for the Housing and Functional Parts

The SHARK is designed to be printed on commercial 3D printers for home and professional use. The printer must be able to process ABS, and have a minimum printing space of 200×200×180 mm (LxWxH). Also, the printing platform should be heated to avoid warping at large parts, or covered with ABS Juice. For the prototypes a Zortrax M200 with FDM (Fused Deposition Modeling) technology is used. For support structures the printer uses the same as the printing material, thus it has to be mechanically removed in postproduction. ABS is mostly preferred for use in engineering due to its strength, flexibility, longer lifespan, machinability and the resistance to high temperature [29] and is therefore used for the development of the SHARK.

a) Impact of FDM on the mechanical properties of ABS: A study on the mechanical properties of parts printed with RepRap low-cost open-source 3D printers found that results of the prints were comparable to those of commercial vendors with average tensile strengths of 28.5 MPa and average elastic moduli of 1807 MPa [19], [30]. Calculations on the functional parts resulted in maximum forces of $F_C =$ 28.9342 N for the buoyancy module and $F_C =$ 7.6226 N for the pitch module assuming a load weight of 1 kg, and are therefore negligibly small.

b) Waterproofing the prints: The 3D printed parts are not initially waterproof due to inaccuracies during the printing process which can affect the surface. Using FDM technology instead of e.g. injection molding, the outer shell of an object may have leaks. Since the material is extruded on a path layer by layer, it is more a lattice structure than a solid one. Also, 3D printers usually do not print solid structures, but shells of about 0.8 mm minimum thickness with filling material. This saves printing time and material costs without significant loss of rigidity but is unfavorable for creating a waterproof print. Another critical point for the surface are contacts to supporting structure on the outside of the shell, since parts of the shell can be ripped out when removing the support material. After joining and machining the parts, the surface has to be treated to be waterproof. To smoothen and seal the surface, methyl ethyl ketone (MEK) or acetone have to be applied vaporized. The substances can be used to polish the surface, or to glue parts together by locally dissolving them. The drawback of using acetone for treatment of the surface is the loss of surface quality as well as the loss of shape accuracy. Alternatively, the prints can be coated with conventional paint. Due to the lattice structure, the paint enters spots where the shell may be damaged and seals it when dried. The drawback of coating is that the process is time consuming, and the results may vary depending on the used paint.

C. Hedgehog Integrated Development Environment

As there currently is no versatile way of developing code on the Hedgehog controller, a web based development environment for the Hedgehog Controller was implemented. This especially improves its usability, keeping educational robotics in mind.

1) RESTful API: Both the IDE frontend and the RESTful API are written using the TypeScript programming language. This allows code sharing between those concepts and follows the DRY principle. Another advantage of the JavaScript and NodeJS [31] platform on the server is good asynchronous and real-time support [32]. The API exposes the following core functions:

- Create, update and delete programs. As every user program is a git repository this part of the API enables a highly simplified but at the same time more beginner friendly view to the repository. This way, previous versions of programs may be accessed as read only and the most recent version is always checked out in the working tree. New versions can be created by saving the working tree state as new version. Moreover, the sole way to reset a program to an existing version is to jump back in version history and consequently delete all subsequent versions.
- Interacting with peripherals of the hedgehog controller. The API server adapts the Hedgehog protocol and allows both access to the controller's actors (motors and servos) and readout of sensors. This feature is especially useful for testing and debugging purpose.
- Executing programs on the controller. Due to the fact that it is not reasonable to execute two or more hardware accessing programs concurrently, only one program can be run at a time. As there might be programs which do not use motors or servos (e.g. for applications with server/client architecture), support for multi program execution is planned for the future.

One main goal of SHARK and the Hedgehog Robotics Controller is to create an open and easily extendable platform. Thus, the RESTful HTTP service follows the JSON API standard, a specification which defines how request and response documents are constructed. The API itself is specified with the Open API Specification [33]. Using the resulting definition file, the API can be easily mocked and developed against, without relying on the actual implementation.

2) IDE frontend: The IDE consists of four main views:

- A program view for managing programing for creating, updating, deleting and executing programs.
- The textual programming environment allows users to write their programs. Currently, only programs written in the Python programming language are supported.
- A visual programming environment for technology-freshmen.
- A motor, servo and sensor control page for rapid testing and debugging.

The IDE is implemented as a single page application [34] using the frontend framework Angular 2 [22]. Single page applications are web applications which do not reload the whole page on every action. Instead, all necessary structural components are initialized on the first page load and subsequent requests solely contain the needed information formatted as JSON data. This schema helps to avoid network overhead and keeps request times low as no server side rendering is performed.

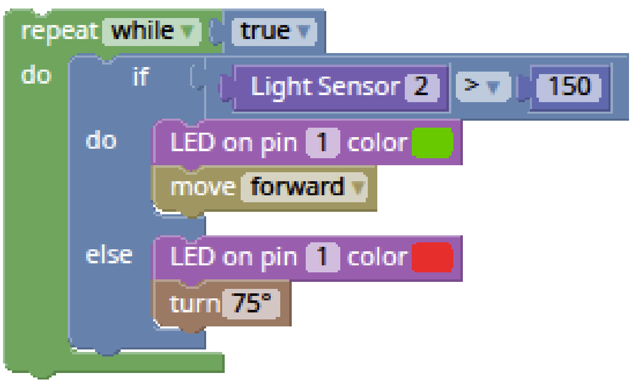


Fig. 6. Blockly hedgehog demo.

a) Textual environment: The textual IDE mainly consists of a control bar for program management, a file tree and a file view for editing the code. The code editor allows for easy programming since it features amongst others autocompletion, syntax highlighting and auto-indentation.

b) Visual programming environment: To make the hedgehog platform even more intuitive for the educational robotics field a visual programming language will be implemented as a way of programming the robot using a block based system. As it would be well beyond the scope of this working to develop a visual programming environment, an existing system called Blockly [35] was embedded into the IDE. It has a convenient way of creating custom blocks for controlling motors and servos or reading sensor values. Blockly itself does not include a runtime for program execution. Instead, it is able to generate code for various languages including Python.

V. Conclusion

Maritime robotics is an important area because of its vast amount of usages. Similarly, educational robotics is a growing field of getting students involved in new studies that they would not have known otherwise. In this paper we presented an opportunity to combine both maritime and educational robotics while keeping it low cost and highly usable. Using robotics as a teaching method requires intuitive programming and a reason to keep young students interested. The SHARK aims to cover both of these points. The basic functionalities of an underwater robot, such as diving, controlling the pitch, moving and steering were reached using a modular design. The 3D printing technology is not simply usable for underwater environment, since the printed parts often have leaky spots on the outer shell where water can enter. Especially the bottom layer of the print, where it is placed on the printing platform, is critical. Waterproof prints can be achieved with lots of effort in the subsequent treatment, such as acetone vaporizing and using coating. On the software side, easy to use development environment for Hedgehog Light robotics controller was presented. Moreover, it has been shown that both the need of beginners and advanced users can be combined into a single application, using modern web technologies. During the overall systems design, special attention has been directed to educational use cases as well as workshop setups and usability at competitions. The SHARK was introduced for the underwater robotics competition at ECER 2016. There were three student groups attending the competition, from age 15 to 17. The main functionality of the robot could be tested successfully in a tank. The teams were able to solve the competition tasks, such as diving down and touching specific objects on the ground of tank, after only one day of workshop and 3 days of supported preparation time. Competition tasks of the following years should address issues of the marine environment, e.g. the pollution with plastics, to encourage the students to raise awareness for the sensitive eco-system and work on conceptual solutions. Due to the focus on using the motors and sensors used in the Botball®Program, the robot's abilities are limited. Especially due to the motors, the size of the robot is considerably large and a lot of load weight has to be used. By using other electronic parts, not only the robot's size could be improved, but also the performance. By implementing mechanical and electrical control systems [36] the response time and energy consumption of the robot can be optimized. The project's aim is and will be to be low-cost, therefore it shall be examined if motors and sensors can be waterproofed with a small budget that are compatible to the used controller. The lessons learned of the prototype of the SHARK, is that the 3D printing technology is only partially suitable for the waterproof housing but very useful for other functional parts. Therefore, in it should be discussed in further work how the performance of the shell can be optimized, e.g. by using another technology than FDM. On software side it can be considered to implement APIs especially designed for underwater robotics applications. The APIs should simplify the robot's use in the complex underwater environment, by implementing functions like depth holding, stability and navigation with control systems engineering in the background. The addressed users of educational robotics should be able to focus on the solving of simple tasks and exploring the underwater environment.

Acknowledgment

The authors would like to thank the KISS Institute for Practical Robotics (KIPR) for establishing the Botball®Educational Robotics Program, which played an essential role in the development of our passion for robotics. Also, we would like to thank the parent's association at the Vienna Institute of Technology (Technologisches Gewerbemuseum, TGM) for their financial support over the last years. Moreover, we would like to thank the Practical Robotics Institute Austria and their staff - especially Clemens Koza and Daniel Frank for working on the Hedgehog controller.

References

[1] Hawai'I, Pacific University Oceanic Institute. Aqua facts. [Online]. Available: http://www.oceanicinstitute.org/aboutoceans/aquafacts.html

[2] National Oceanic and Atmospheric Administration, National ocean service. [Online]. Available: http://oceanservice.noaa.gov/facts/exploration.html

[3] New Scientist, "How many species live in the sea?" [Online]. Available: https://www.newscientist.com/article/dn14206-how-many-species-live-in-the-sea/

[4] Discovery News, "Two-thirds marine species remain unknown," [Online]. Available: http://news.discovery.com/animals/whales-dolphins/marine-species-unknown-121115.htm

[5] National Geographic, "Save the plankton, breathe freely," [Online]. Available: http://education.nationalgeographic.org/activity/save-the-plankton-breathe-freely/

[6] Y. He, S. Guo, L. Shi, S. Pan, and Z. Wang, "3d printing technology-based an amphibious spherical robot," in *Proc. 2014 IEEE International Conference on Mechatronics and Automation*, Aug. 2014, pp. 1382– 1387.

[7] B. P. P. Chotikarn, W. Koedsin and P. Aiyarak, "Low cost submarine robot," Songklanakar, *Journal of Science and Technology*, vol. 32, no. 5, pp. 513–518, 2010.

[8] N. Koenig and M. Mataric´, *Motorola Semiconductor Data Manual*, Motorola Semiconductor Products Inc., Phoenix, AZ, 1989.

[9] F. Seifer, *Materials for Enabling Hands-on Robotics and Stem Education*, in AAAI Spring Symposium on Robots and Robot Venues, Stanford, CA: Resources for AI Education, March 2007.

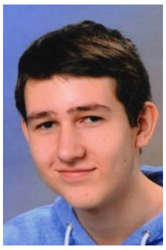
[10] C. Krofitsch, C. Hinger, M. Merdan, and G. Koppensteiner, "Smartphone driven control of robots for education and research," in *Proc. 2013 IEEE International Conference on Robotics, Biomimetics, and* Intelligent *Computational Systems (ROBIONETICS), IEEE*, 2013, pp. 148–154.

[11] A. Eguchi, "Robotics as a learning tool for educational transformation," in *Proc. 4th International Workshop Teaching Robotics, Teaching*

Robotics & 5th International Conference Robotics in Education, 2014, pp. 27–34.
[12] A. M. Ortiz, B. Bos, and S. Smith, "The power of educational robotics as an integrated stem learning experience in teacher preparation programs," *Journal of College Science Teaching*, vol. 44, no. 5, 2015.
[13] H. Goh and B. Aris, "Using robotics in education: lessons learned and learning experiences," *Smart Teaching & Learning: Re-engineering ID, Utilization and Innovation of Technology*, vol. 2, 2007.
[14] S. Ziaeefard, G. Ribeiro and N. Mahmoudian, "Guppie, underwater 3d printed robot a game changer in control design education," in *Proc. American Control Conference (ACC)*, July 2015, pp. 2789–2794.
[15] B. Mitchell, E. Wilkening, and N. Mahmoudian, "Developing an underwater glider for educational purposes," in *Proc. 2013 IEEE International Conference on Robotics and Automation (ICRA)*, May 2013, pp. 3423– 3428.
[16] Pelican Products, Inc., "Pelican micro case," 1020. [Online]. Available: http://www.pelican.com/us/en/product/watertight-protector-hard-cases/micro-case/standard/1020/
[17] B. Barker, "Robots in K-12 education: A new technology for learning: A new technology for learning, ser. advances in early childhood and K-12 education," *Information Science Reference*, (2012). [Online]. Available: https://books.google.at/books?id=DDnH8uoEYlIC
[18] X. Che, I. Wells, G. Dickers, P. Kear, and X. Gong, "Re-evaluation of rf electromagnetic communication in underwater sensor networks," *Communications Magazine, IEEE*, vol. 48, no. 12, pp. 143–151, December 2010.
[19] J. A. Curcio and C. C. Petty, "The near infrared absorption spectrum of liquid water," *JOSA*, vol. 41, no. 5, pp. 302–304, 1951.
[20] R. Grabler, "Active protective-Housing for use in maritime educational robotics," 2016.
[21] Practical Robotics Institute Austria (PRIA), "Hedgehog light hardware controller," [Online]. Available: http://www.pria.at
[22] C. Krofitsch, W. Lepuschitz, M. Klein, and G. Koppensteiner, "Flexible development environment for educational robotics," in *Proc. International Conference on Control, Automation and Robotics (ICCAR). IEEE*, 2015, pp. 1–7.
[23] P. Deeleman, "Learning angular 2," *Packt Publishing Ltd.*, 2016.
[24] M. Jakl, *Representational State Transfer*, 2005.
[25] P. Hintjens, "Zero MQ: Messaging for many applications," *O'Reilly Media, Inc.*, 2013.
[26] C. Sun and C. Ellis, "Operational transformation in real-time group editors: issues, algorithms, and achievements," in *Proc. the 1998 ACM Conference on Computer Supported Cooperative Work, ACM*, 1998, pp. 59–68.
[27] I. S. V. Pshikhopov, Boris Gurenko and M. Beresnev, "Development and research of path-planning module for control system of underwater vehicle," *International Journal of Mechanical Engineering and Robotics Research*, vol. 5, no. 4, October 2016.
[28] Krueger, "Stefan. Schiffspropeller," [Online]. Available: http://www.ssi.tu-harburg.de/doc/webseiten_dokumente/ssi/vorlesungsunterlagen/propeller.pdf
[29] H. C. Kuhlmann, *Strömungsmechanik. Pearson Studium*, 2007.
[30] ProtoParadigm. "The difference between abs and pla for 3d printing," [Online]. Available: http://www.protoparadigm.com/news-updates/the-difference-between-abs-and-pla-for-3d-printing/
[31] B. Tymrak, M. Kreiger, and J. Pearce, "Mechanical properties of components fabricated with open-source 3-d printers under realistic environmental conditions," *Materials & Design*, 58, pp. 242 – 246, (2014). [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0261306914001538
[32] D. Herron, "Node. js web development," *Packt Publishing Ltd.*, 2016.
[33] S. M. Zhao, X. L. Xia, and J. J. Le, "Areal–time web application solution based on node.js and web socket," in *Advanced Materials Research, Trans Tech Publ.*, vol. 816, pp. 1111–1115, 2013.
[34] B. Varanasiand and S. Belida, "Documenting rest services" in *Spring REST.*, Springer, pp. 91–104, 2015.
[35] M. S. Mikowski and J. C. Powell, "Single page web applications," *B and W*, 2013.
[36] A. Marron, G. Weiss, and G. Wiener, "A decentralized approach for programming interactive applications with javascript and blockly," *Proceedings of the 2nd edition on Programming systems, languages and applications based on actors, agents, and decentralized control abstractions. ACM*, pp. 59–70, 2012.
[37] A. Jebelli and M. C. E. Yagoub, "Fuzzy logic pid based control design for a small underwater robot with minimum energy consumption," *International Journal of Mechanical Engineering and Robotics Research*, vol. 5, no. 3, July 2016.

Reinhard Grabler was born in 1991 in Korneuburg. After graduation at the TGM (Vienna Institute of Technology) in 2010 and after completing the military service he started his bachelor studies in mechanical engineering - management on Vienna University of Technology. From October 2011 to June 2013 he has been a research assistant at the Institute of Automation and Control at Vienna University of Technology. Since the founding of PRIA in April 2012, which he was part of, he has been working as a research assistant there. Ever since he has been a member of the organization team and the conference committee of the European Conference on Educational Robotics (ECER).

Markus Klein was born 1998 in Vienna, Austria. He is currently in his final year of the computer science department at TGM (Vienna Institute of Technology) and working on the software platform of the Hedgehog Controller as part of his diploma project. In 2014 has won the national skills competition of mobile robotics and successfully attended the world skills competition in Sao Paulo, Brazil in 2015. Also, since 2013, he is member of PRIA and since 2015 he is a part of the organization team of the European Conference on Educational Robotics (ECER). Besides school, he is also working as a part-time software developer for a start-up company.



Thomas Fellner was born in Vienna, Austria, in 1998. He got engaged with robotics when he started studying at the TGM in the computer science department and now he is finishing his high school education while writing a diploma thesis about further developing the Hedgehog platform as part of the final exam. He took part in the European Conference on Educational Robotics (ECER) 2015 competing in the aerial tournament finishing in third place.

Gottfried Koppensteiner was born in Vienna in 1979. After graduation at the TGM in 1999 (Electronics, Biomedical Engineering Education Branch) he operated as an assistant and is now the head of the department of computer science there. He graduated 2005 as a Dipl.-Ing. (FH) and continued his studies with a master degree at the Vienna University of Technology in computer science and graduated with his work, "Design and development of an ontology for a knowledge-based automation system" in May 2007. He has finished his PhD about "Knowledge-Based Agent Architecture for Cooperative Production Systems" in Fall 2012 with distinction and his current research focus is robotics for industry and education. Since April 2012 he is chairman of PRIA.