HardwareX 6 (2019) e00076

Contents lists available at ScienceDirect

HardwareX

journal homepage: www.elsevier.com/locate/ohx

A low-cost and small USV platform for water quality monitoring

Wonse Jo [a], Yuta Hoashi [a,b], Lizbeth Leonor Paredes Aguilar [c], Mauricio Postigo-Malaga [c], José M. Garcia-Bravo [d], Byung-Cheol Min [a,*]

[a] *SMART Laboratory, Department of Computer and Information Technology, Purdue University, West Lafayette, IN 47907, USA*
[b] *School of Mechanical Engineering, Purdue University, West Lafayette, IN 47907, USA*
[c] *School of Electronics Engineering, Universidad Nacional de San Agustín, Arequipa, Peru*
[d] *School of Engineering Technology, Purdue University, West Lafayette, IN 47907, USA*

ARTICLE INFO

Keywords:
Unmanned Surface Vehicles (USVs)
Water Monitoring System
Underwater Thruster Control
Arduino

ABSTRACT

We propose a new, fully open-source, low-cost, and small-sized Unmanned Surface Vehicle (USV) for measuring near-surface water quality in real time in various environments. Existing commercial USVs are expensive and not fully based on open-source hardware, making it difficult to purchase them and to modify their designs and programming to suit various environments. In contrast, the USV platform proposed in this paper is completely open-source, from hardware to software; most parts of the platform can be 3D-printed, and it can be easily modified and upgraded in terms of both design and programming. The platform is equipped with off-the-shelf water sensors for acquiring data like as pH, turbidity, and temperature to measure water quality in real water resources (such as ponds, reservoirs, and lakes). Furthermore, we provide an Android application through which users can easily control the USV via Bluetooth and display the sensor and GPS data the platform generates. We validated the performance of the platform in terms of usability, mobility, and stability through field experiments in various locations, including both the USA and Peru. Moreover, we present a potential application and approach in which this platform can navigate autonomously by utilizing the Robot Operating System (ROS) and Bluetooth protocols.

Specifications table:

Hardware name	SMARTBoat 3
Subject area	Environmental Robotics Ocean Robotics
Hardware type	Unmanned Surface Vehicles (USVs) Water Quality Monitoring Field Measurements and Sensors
Open source license	GNU GPL v3
Cost of hardware	$ 200.97
Source file repository	Open Science Framework: https://osf.io/wsnrt/

* Corresponding author.
E-mail addresses: jow@purdue.edu (W. Jo), yhoashi@purdue.edu (Y. Hoashi), lparedesa@unsa.edu.pe (L.L. Paredes Aguilar), mpostigom@unsa.edu.pe (M. Postigo-Malaga), jmgarcia@purdue.edu (J.M. Garcia-Bravo), minb@purdue.edu (B.-C. Min).

https://doi.org/10.1016/j.ohx.2019.e00076
2468-0672/

1. Hardware in context

Water quality is a large environmental problem and one of humanity's grand challenges. For example, with less than 4 mm/year of rain the southwestern region of Peru suffers scarce water resources that have been contaminated by the mining industry [1]. Most countries in the world have been confronted with water pollution caused by oil spills, plastic waste, and human activities. Such contamination can damage the habitats of fish and other aquatic life, agriculture, and ultimately can be detrimental to human health. On top of this, delayed response to a water crisis can incur tremendous costs, require considerable recovery time, and provoke social and political strife. One example of water quality issues is presented in Lake Erie [2] and Lake Taihu [3] where toxic algal blooms are contaminating drinking water reservoirs. Therefore, it is crucial to monitor and track water condition in order to ensure the safety and quality of watersheds and safeguard water resources from contamination.

Currently, water quality monitoring can be done manually by a human but also autonomously, frequently, and efficiently through the application of robotic systems [4–6] and various USV platforms [7]. For example, a stationary, submerged water monitoring device called the Environmental Sample Processor was introduced in 2009 [8]. This device monitors water quality regularly and wirelessly sends the results to researchers. Another monitoring device is the AutoNaut Automated Sea Vehicle, an autonomous boat that harvests energy from solar power and the natural pitching and rolling of the sea [9]. It communicates with satellites for receiving instructions and for sending the data it gathers. Additionally, the Platypus is an autonomous commercial robotic watercraft with a polyurethane hull that incorporates sensors to measure dissolved oxygen, temperature, conductivity, and pH [10]. Mounting a smartphone to the vehicle allows it to send data and receive instructions. The Platypus is mainly used for environmental monitoring, flood response, and fish farming. A floating pillar-shaped device has been developed by the University of California, Berkeley [11]; it has bottom-mounted sensors and sends sensor information wirelessly. It can locate itself with GPS and moves around by means of small propellers powered by a Li-ion battery. Kaizu et al. developed an unmanned air-boat that is a hovercraft type for mapping water quality included temperature, pH level, dissolved oxygen, electrical conductivity, turbidity and chlorophyll data [12].

While many platforms and devices have been introduced for water quality monitoring purposes, as presented above, the existing platforms are too expensive and too large for individuals to purchase and use in real-world application. Moreover, no wholly open-source USV platforms are available that would allow customization for varied purposes. Therefore, we propose a new, fully open-source–based, small and low-cost USV platform capable of measuring water quality in real time. Fig. 1 illustrates a summary of the proposed USV platform.

2. Hardware description-SMARTBoat 3

We have developed a new, fully open-source USV platform named the SMARTBoat 3 that is equipped with water quality sensors as shown in Fig. 2. Representative features of the platform are:

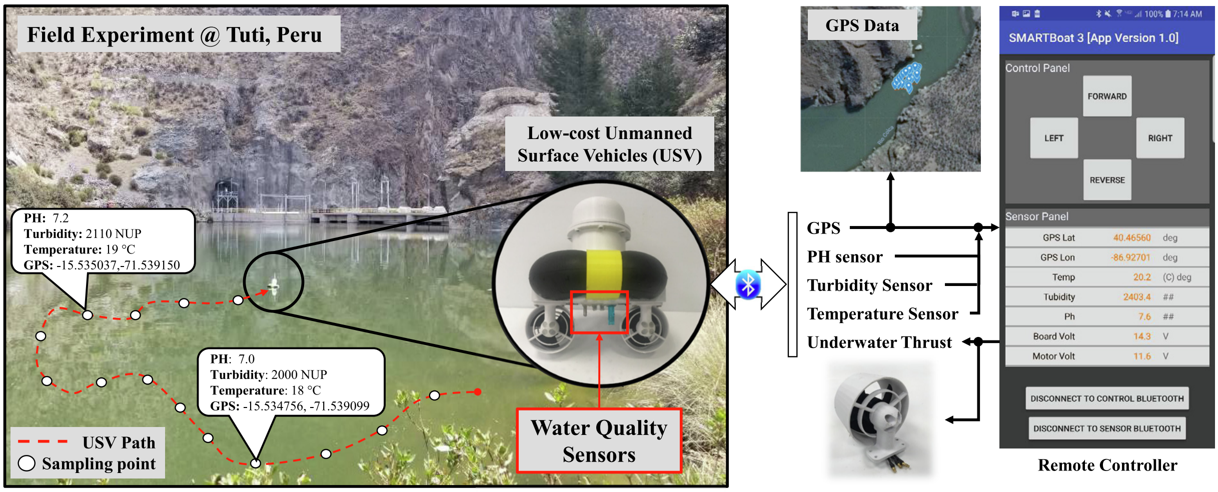
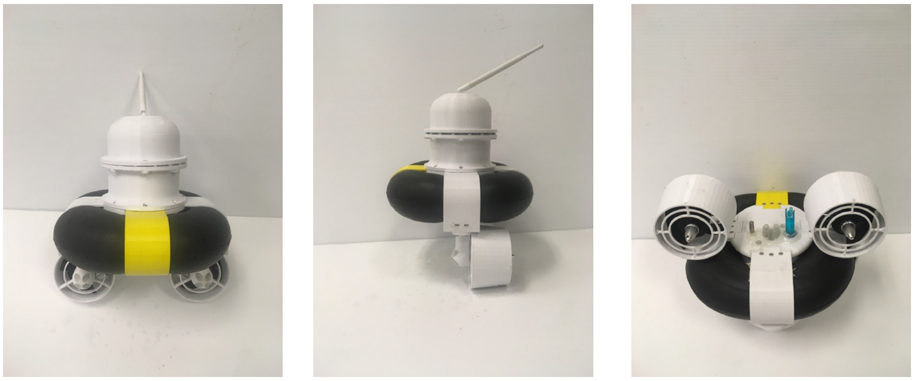


Fig. 1. A proposed USV platform, called SMARTBoat 3, capable of measuring water quality in real time.



(a) (b) (c)

Fig. 2. SMARTBoat 3 platform: (a) front view, (b) side view, and (c) bottom view.

- Low cost and fully open-source-based USV platform
- Available to monitor water quality in real time
- Easy to assemble and modify all resources
- Easily controlled by smartphone application, or Bluetooth devices.

To the best of our knowledge, this paper is the first to present a USV platform that is completely open-source, from hardware design and electronic circuits to program source code and the controlling Android application. Although the design of the platform is inspired by the JALC Boat [13], we have significantly modified most parts to suit our own purposes and to mount water quality sensors. The platform proposed in this paper is smaller in size and lighter in weight than other commercial USVs. Also, it is very low-cost because most parts are 3D-printed, except electronic parts and buoyancy elements.

The dimensions of the platform are 28 × 24 × 28 cm (width × length × height), and its weight is approximately 3 kg, allowing portability and accessibility for experiments/monitoring in inaccessible environments such as those enclosed by high mountains, deep water, and so on. Its maximum speed is approximately 40 cm/s. With all functions operating fully at normal environmental temperature ranging from 20°C to 25°C, including moving at maximum speed and collecting data from these sensors, the estimated operating time is approximately 30 min. In practical applications, assuming that the platform does not always move at maximum speed with both thrusters active, the operating time will be much extended. This platform uses HC-06 as its Bluetooth module, and thus it has a maximum communication range of up to approximately 20 meters [14]. It is capable of measuring water quality in real-world environments using pH, turbidity, and temperature sensors installed on the bottom of the platform (Fig. 2c). The SMARTBoat 3 should be operated in environments where the wind speed is less than approximately 4m/s and wave heights and lengths are less than approximately 7 cm and 165 cm, respectively. These environmental limitations are due to the size of the boat, and were calculated based on the literature presented in [15,16]. We also provide a smartphone application for users to easily and directly control the USV platform via their own smartphones, removing the need to purchase special remote controllers such as a radio-control remote controller. Fig. 3a shows details for each part of the platform. In the lower section are two thrusters operated by Brushless Direct Current (BLDC) motors and an inner tube that provides buoyancy and is sufficiently hard to prevent punctures created by the thorns of aquatic plants and waste material floating on the surface; the sensors for water quality monitoring are located below the inner tube that provides a enough buoyancy force of about 28.6 N. At the top of the inner tube is a hermetic case that contains the electronics blocks of the sensors, power system, actuators, main processing system, and the GPS and Bluetooth modules.

Fig. 3b shows the system architecture of the SMARTBoat 3, which is based on the 8-bit, 16 MHz Atmel ATmega 328p microprocessor. The power block, shown in the upper part of the figure, consists of the 3–4 cell lithium polymer battery that provides 11.1–14.8 volts to feed two thrusters (M1 and M2) and the microprocessors. Also, a voltage regulator of 5 volts to power two microprocessors and the voltage and current sensors for monitoring the current battery power level are presented. The acquisition block for the water quality sensors, presented at the bottom of the figure, has extra sensor pins allowing users to expand the platform with other sensors. The sensors gather data at 10 Hz sampling intervals. The communications block is presented on the left of the figure and consists of a Bluetooth interface to communicate with smart devices, a serial connection RS-232, a GPS device, and a connector for programming.

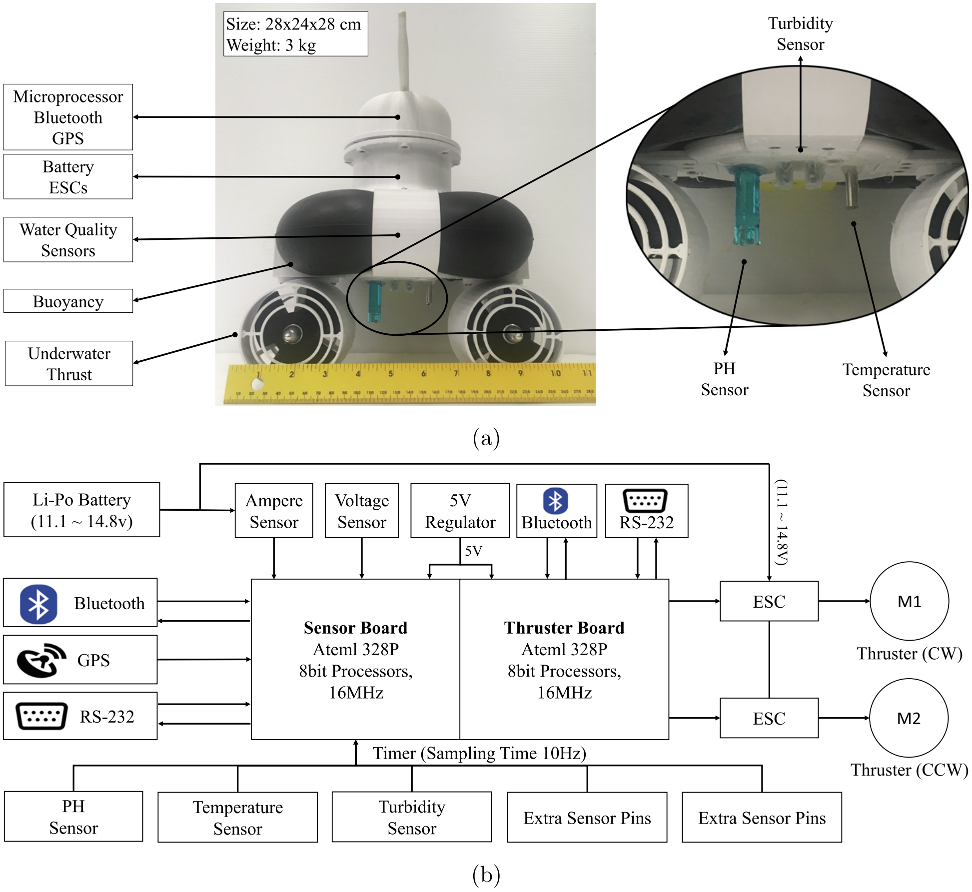


Fig. 3. Detailed specifications of SMARTBoat 3 platform: (a) overall hardware parts installed on the platform, and (b) system architecture of SMARTBoat 3.

Regarding Bluetooth communication, the SMARTBoat 3 has two Bluetooth communication modules; one is for receiving each motor value from a smartphone application, and the other is for transmitting sensor data to the operator's smartphone. The Bluetooth protocols for transmitting/receiving communication data are presented in Tables 1 and 2, respectively. Both protocols use a semicolon (;) as the delimiter between data elements.

Table 1
Bluetooth protocol of GPS and sensors.

Type	GPS Data		Water Quality Sensor Data			Battery for Boards		Battery for Thruster	
Detail	Longitude ;	Latitude ;	Temperature ;	Turbidity ;	pH level ;	Voltage ;	Ampere ;	Voltage ;	Ampere ;
Unit	integer	integer	°C	NTU	0 to 14 pH	V	A	V	A

Table 2
The Bluetooth protocol to control underwater thruster.

Type	Left Thruster (CCW)	Right Thruster (CW)
Detail	−100 to 100 ;	−100 ;
Unit	integer	integer

3. Design files

3.1. Hardware files

Design filename	File type	Open source license	Location of the file
battery_ESC_layer.STL	STL & Solidworks	GPL 3.0	https://osf.io/sqmka/
box_holder_arm.STL	STL & Solidworks	GPL 3.0	https://osf.io/bnwpa/
controller_hold_layer.STL	STL & Solidworks	GPL 3.0	https://osf.io/gf84k/
extender_control_box.STL	STL & Solidworks	GPL 3.0	https://osf.io/a7pgb/
main_control_box_cap.STL	STL & Solidworks	GPL 3.0	https://osf.io/3vce7/
propeller_blade_3_CCW.STL	STL & Solidworks	GPL 3.0	https://osf.io/d7q3n/
propeller_blade_3_CW.STL	STL & Solidworks	GPL 3.0	https://osf.io/7p8ny/
propeller_protector.STL	STL & Solidworks	GPL 3.0	https://osf.io/wa7x5/
rope_holder.STL	STL & Solidworks	GPL 3.0	https://osf.io/674gc/
sensor_box.STL	STL & Solidworks	GPL 3.0	https://osf.io/4uejr/
sensor_hold_layer.STL	STL & Solidworks	GPL 3.0	https://osf.io/kfujw/
sensor_module_holder.STL	STL & Solidworks	GPL 3.0	https://osf.io/crbw4/
sensor_protect_cover.STL	STL & Solidworks	GPL 3.0	https://osf.io/6n4bp/
Thrust_holder_arm.STL	STL & Solidworks	GPL 3.0	https://osf.io/vqh9e/
Thrust_Protector.STL	STL & Solidworks	GPL 3.0	https://osf.io/x3j84/
Thrust_holder.STL	STL & Solidworks	GPL 3.0	https://osf.io/k986u/
Sensor_and_GPS_board.pdf	PDF	GPL 3.0	https://osf.io/wb63t/
Thrust_control_board.pdf	PDF	GPL 3.0	https://osf.io/6xa83/

- battery_ESC_layer: a layer of covers to hold batteries and ESCs (required 1 ea).
- box_holder_arm.STL: to connect and hold a main control box (called a sensor bax.STL) with the inner tube (required 2 ea).
- controller_hold_layer.STL: a layer of covers to contain the Arduino boards and GPS module (required 1 ea).
- extender_control_box.STL: extends the main control box to contain the battery, sensor modules, and wires (required 1 ea).
- main_control_box_cap.STL: a cap to protect the main control box and prevent the Bluetooth antenna from splashing water (required 1 ea).
- propeller_blade_3_CCW.STL: 3D-printed counterclockwise propeller mounted on the left side of the platform (required 1 ea).
- propeller_blade_3_CW.STL: 3D-printed clockwise propeller mounted on the right side of the platform (required 1 ea).
- propeller_protector.STL: a protector case to protects the propeller from getting tangled with aquatic plants (required 2 ea).
- rope_holder.STL: (optional) a rope holder to drag the platform the from the boat in emergency situations (required 1 ea).
- sensor_box.STL: a fundamental control box used to build other boxes; contains a battery, ESCs, and sensors measuring water pH, turbidity, and temperature (required 1 ea).
- sensor_hold_layer.STL: a layer part to support the middle of the pH and temperature sensors (required 1 ea).
- sensor_module_holder.STL: a layer part to fix sensor ADC modules (required 1 ea).
- sensor_protect_cover.STL: (optional) an exterior cover box to protect sensors from the environment and unintended damage (required 1 ea)
- Thrust_holder_arm.STL: a connector to hold a thruster with the main control box (required 2 ea).
- Thrust_Protector.STL: a ducted propeller case to improve the efficient of the propeller and protect the thruster from being tangled by aquatic plants (required 2 ea).
- Thrust_holder.STL: assembles and connects a propeller to the main box (required 2 ea).
- Sensor_and_GPS_board.pdf: electronic board circuits for the transmission of data from water quality sensors and GPS via Bluetooth.
- Thrust_control_board.pdf: electronic board circuits for controlling thrusters (CW/CCW) via Bluetooth.

3.2. Software files

Design filename	File type	Open source license	Location of the file
control_code_esc.ino	Arduino Sketch	GPL 3.0	https://osf.io/km73c/
control_code_sensor_GPS.ino	Arduino Sketch	GPL 3.0	https://osf.io/gejnd/
SMARTBoat 3_remote_controller.apk	APK	GPL 3.0	https://osf.io/qfv8k/

- control_code_esc.ino: Arduino sketch for the thruster control board that controls both CW (clock wise) and CCW (counter-clock wise) thrusters through an electronic speed control (ESC).
- control_code_sensor_GPS.ino: Arduino sketch for the sensor and GPS board, which read sensor and GPS data.
- android program: Android app to control the USV and display data from water sensors and GPS at 10 Hz sampling intervals. The app allows saving all data on the smartphone.

4. Bill of materials

- BOM_SMARTBoat-03:https://osf.io/9enuw/

Designator	Component	Number	Cost per unit currency	Total cost	Source ofmaterials	Material type
Microcontroller	Adafruit Pro Trinket	2	$ 9.95	$ 19.90	https://www.adafruit.com/product/2000	Others
Battery	Turnigy2200mAh 4S30C Lipo Pack	1	$ 25.54	$ 25.54	https://hobbyking.com/en_us/turnigy-2200mah-4s-30c-lipo-pack.html	Others
Brushless Motor	PROPDRIVEv2 2826 1000KV Brushless Outrunner Motor	2	$ 15.44	$ 30.88	https://hobbyking.com/en_us/propdrive-v2-2826-1000kv-brushless-outrunner-motor.html	Others
ESC	HobbyKing 30A ESC 3A UBEC	2	$ 11.41	$ 22.82	https://hobbyking.com/en_us/hobby-king-30a-esc-3a-ubec.html	Others
Buoyancy	Farm & Ranch 10" Replacement Tire Inner Tube	1	$ 5.99	$ 5.99	https://www.menards.com/main/tools-hardware/automotive/tires-tire-care/farm-ranch-10-replacement-tire-inner-tube/fr2300/p-1444445219843.htm	Others
pH sensor	Gravity: Analog pH Sensor/ Meter Kit For Arduino	1	$ 29.50	$ 29.50	https://www.dfrobot.com/product-1025.html	Others
Turbidity Sensor	Gravity: Analog Turbidity Sensor For Arduino	1	$ 9.90	$ 9.90	https://www.dfrobot.com/product-1394.html	Others
Temperature Sensor	Gravity: Waterproof DS18B20 Sensor Kit	1	$ 7.50	$ 7.50	https://www.dfrobot.com/product-1354.html	Others
GPS	EM506	1	$ 39.95	$ 39.95	https://www.sparkfun.com/products/12751	Others
Bluetooth	HC-06	2	$ 4.50	$ 8.99	https://www.amazon.com/dp/B07BRMLBDF	Others

5. Build instructions

5.1. Firmware update of commercial ESCs

Before beginning to assemble the platform, it is important to install a new, open-source-based firmware for the Atmega-based 3-phase sensor-less motor electronic speed control (ESC). This is because most ESCs for Unmanned Aerial Vehicles (UAVs) (e.g. drones) support only one directional motor, either clockwise or counterclockwise. However, our USV should use a bi-directional mode to drive back and forth, which requires replacing the installed firmware with SimonK's firmware [17]. It is easy to upload this firmware using Arduino Uno through the instruction document provided on the SimonK website.

5.2. Configuration of electronic circuits

In order to prevent unexpected system errors during communication, the boat has two main Arduino boards, both using the Adafruit Pro trinket that is based on the ATmega328 at 5 V, 16 MHz: (1) the sensor and GPS board, and (2) the thruster

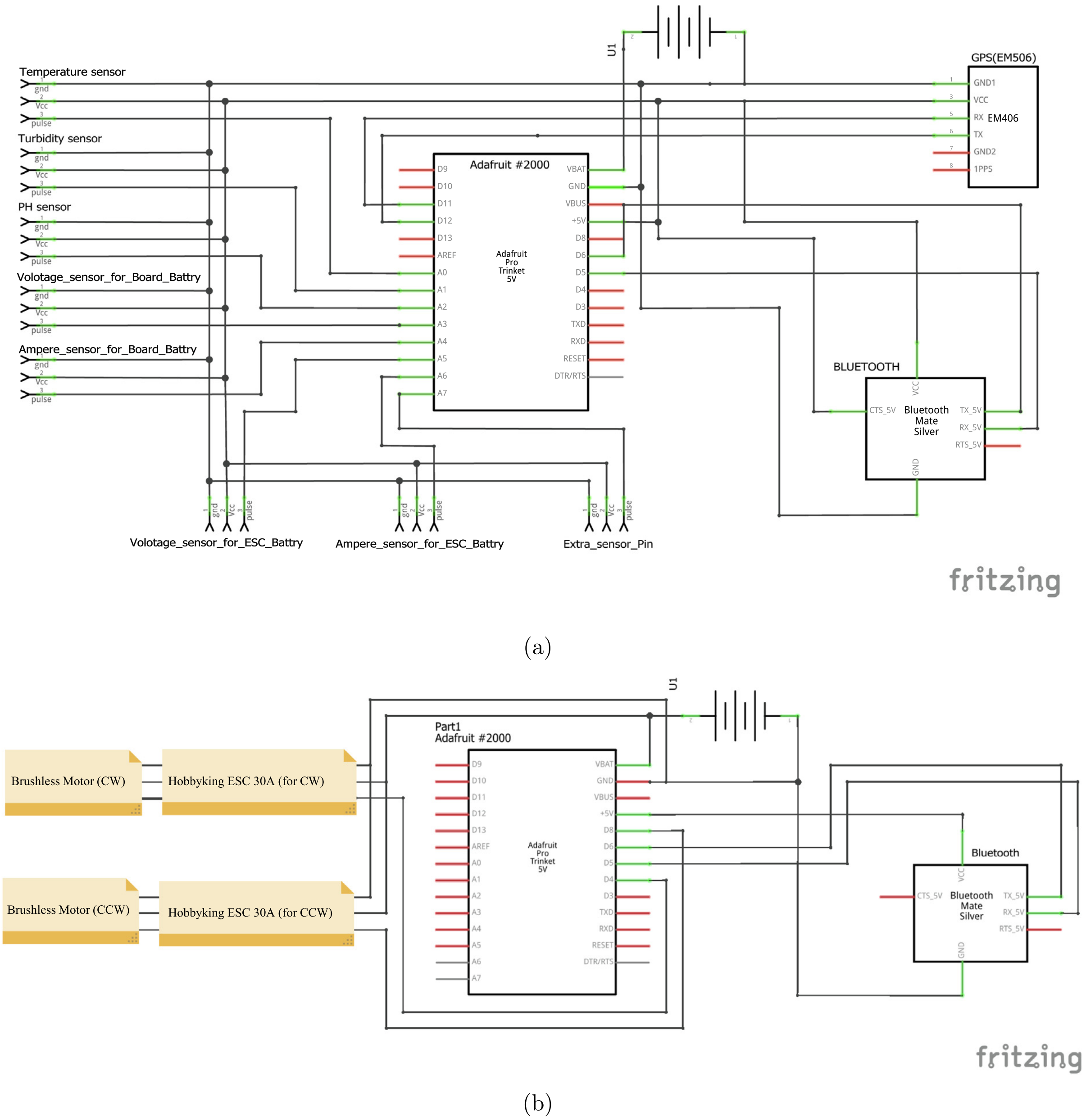


Fig. 4. Electronic circuit schematic: (a) sensor and GPS board, and (b) thrust control board.

control board. The sensor and GPS board (Fig. 4a) collects data from the water quality sensors and GPS at 10 Hz sampling intervals, and sends them to a smartphone application via Bluetooth. The thruster control board (Fig. 4b) is focused on controlling the two thrusters according to commands received from the smartphone application. Circuit schematic diagrams for both Arduino boards are available at these corresponding open-source websites:

- Sensor and GPS board: https://osf.io/wb63t/
- Thrust control board: https://osf.io/6xa83/

5.3. Assembly instructions for the SMARTBoat 3

The SMARTBoat 3 platform consists of three layers to contain the water quality sensors, ESCs, battery, GPS, Bluetooth, and main boards as shown in Fig. 5. The first layer (main control box) is a fundamental structure to mount the water quality sensors and tightly fix the thrusters (CW/CCW). The second layer adds support to help stabilize the sensors; contains wires,

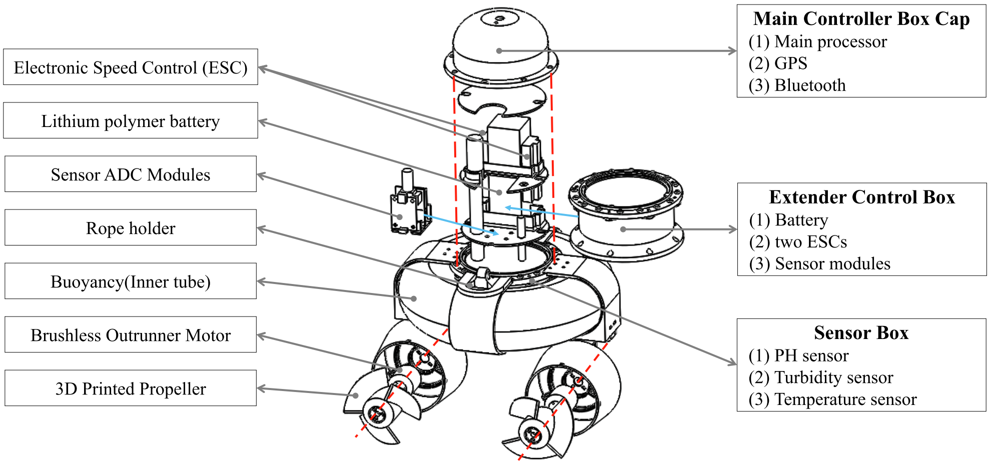


Fig. 5. SMARTBoat 3 boat details as each layer.

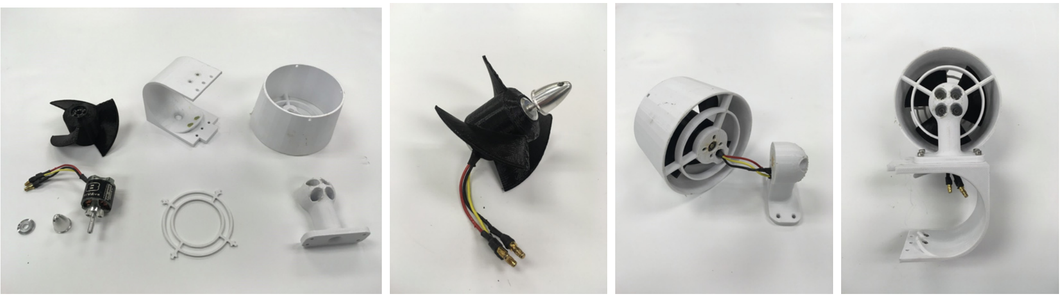
ESCs, and the battery; and protects the platform from unexpected water permeation through slight gaps that exist between the first layer and the sensor module. The third layer contains the Arduino boards, GPS, and Bluetooth modules for the easy reception and transmission of data. The propeller case of the thruster is manufactured using a LULZBOT TAZ 6 3D printer [18] using PLA filament made by eSUN [19]. The SMARTBoat 3 is printed with PLA filament, which is widely used in 3D printers for reason of its relatively low cost and ease of printing. If we want to make the platform more robust and extend its life, we can use other types of filaments such as ABS or PETG. It is worth noting that all parts uploaded to the repository should be printed out before starting assembly. As illustrated in Figs. 6 and 7, assembly of all parts is required to make the complete platform.

5.3.1. Instructions for propeller assembly

In the SMARTBoat 3, there are two underwater thrusters each using a BLDC motor. Our prototype uses outrunner motors, which are a type of brushless DC electric motor that is widely used in unmanned aerial vehicles (e.g., drones). The motor spins its outer shell around its case, and thus there is no short circuit directly caused by water. Also, Blue Robotics, one of the leading companies building high-performance marine robotics components and systems, utilizes this type of motor in a remotely operated underwater robot (e.g., unmanned submarine) [20]. Finally, many researchers use BLDC motors in their underwater research [21,22]. Fig. 6 shows the procedure for assembling the propeller. Fig. 6a summarizes the overall parts needed to assemble the thruster. Fig. 6b shows the assembly of a BLDC motor with a printed propeller, and how to fasten it with the propeller adapter included in the BLDC motor package. Fig. 6c shows how to insert the wires of the BLDC motor into the thrust holder, and how to fix both using screws. Fig. 6d shows a completely assembled thruster.

5.3.2. Instructions for assembly of the main box and holding arms

Fig. 7 presents assembly instructions for the main container, including sensors, mounting propellers, the battery, ESCs, and the main microprocessors. Fig. 7a details how to mount the pH, turbidity, and temperature sensors on the bottom of the main box. Fig. 7b shows how to add the battery and ESCs, and then attach the second layer to the main box in order



(a) (b) (c) (d)

Fig. 6. Assembly procedures of propeller: (a) fabricate the 3D printed components for building a propeller assembly , (b) assemble the BLDC motor and the propeller by fastening a propeller adapter, (c) attach the propeller on thrust holder part, and (d) add thrust holder to connect it with an arm holder part.

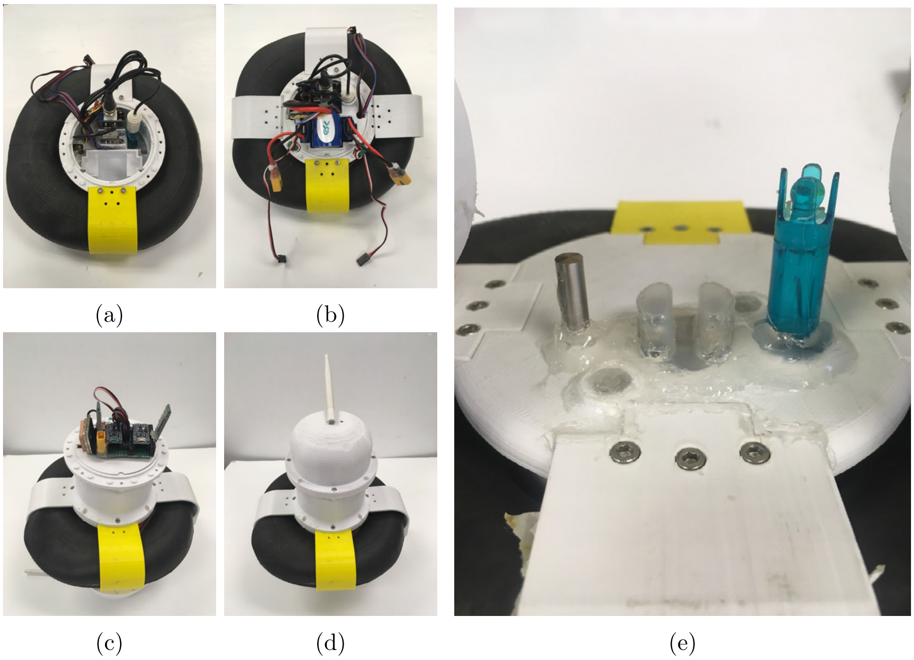


Fig. 7. Assembly procedures of the main box: (a) mount sensors on the main control box, (b) add a battery and ESCs and a second layer to hold the middle part of sensors, (c) extend the control box and holding microprocessors, (d) cover with a cap part, and (e) waterproofed section.

to support the middles of the sensor sticks and enhance waterproofing for the sensor modules and main microprocessors. Fig. 7c presents how to extend the main control box and then add the controller layer and the microprocessor modules. Fig. 7d shows the cap used to cover the whole platform and protect the main processor from water splashes during operation. The last step is to glue around sensor module projections on the bottom of the platform for waterproofing, as shown in Fig. 7e. This helps protect from unexpected water permeation through gaps between the sensors and the main control box.

6. Operation instructions

6.1. Calibration of water quality sensors

The SMARTBoat 3 has three water quality sensors: (1) temperature, (2) pH, and (3) turbidity. The temperature and turbidity sensors are popular models with calibration equations provided by their manufacturers. In contrast, the pH sensor uses the two-point calibration method, so calibration using the two standard solutions (4.0 and 7.0 pH) is necessary for obtaining accurate data. Therefore, before beginning to measure pH values, users are advised to implement the following calibration procedure:

- pH sensor: *Gravity: Analog pH Sensor/ Meter Kit For Arduino*[23]

1. Download sample code for sensor calibration from the below website: https://wiki.dfrobot.com/PH_meter_SKU__SEN0161_
2. Upload the sample code to the Arduino controller.
3. Immerse the electrode in the standard solution with pH 7.0.
4. Open the serial monitor and find the pH value measured by the sensor.
5. Update the “offset” value in the sample code as needed to correct the measurement. For example, if measured value is 6.88, the offset value should be 0.12.
6. Immerse the electrode in the standard solution with pH 4.0, and wait about one minute.
7. Adjust the gain potential devices to stabilize at around pH 4.0.

6.2. Calibration of underwater thrusters

Due to inherent characteristics of the hardware, drift (i.e., a tendency to pull to the right or left) may occur during USV operation. When drift occurs, both underwater thrusters require calibration through a visual inspection. In particular, in order to make the boat drive straight, operators should tune the weight of the motor control values in the Arduino *control_code_esc.ino*. The relevant variables are *weight_CW* and *weight_CCW*, which are float values between −1.0 and 1.0. For example, if the SMARTBot 3 tends to pull to the right, *weight_CW* should be increased or *weight_CCW* decreased.

6.3. Instruction for using the Android application

We developed an application to control the SMARTBoat 3 using an Android-based smart device (e.g. smartphone or tablet) as shown in Fig. 8. The minimum SDK version is 21, and the target SDK version is 26. Control of the platform is easy and direct, as the application interface is similar to typical remote controllers. Video instructions are provided in the repository. The application package and instructional video are available at the links below:

- SMARTBoat-03_remote_controller.apk:https://osf.io/qfv8k/
- Android_Application_instruction.mp4:https://osf.io/3acdn/

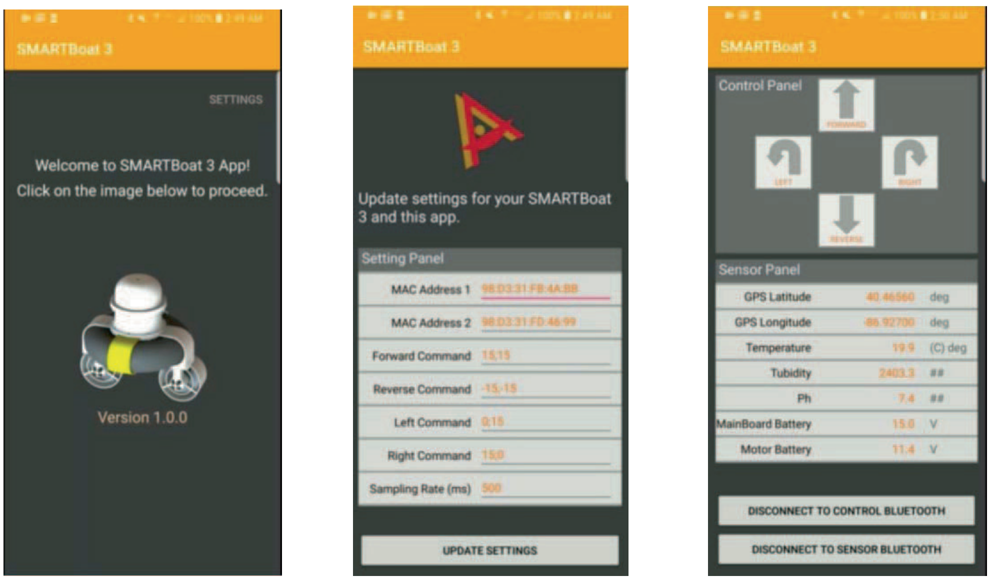
6.4. Operating procedure

1. Connect battery to power jacks:
   (1.1) main power of control system (ESC control board and sensor reading board)
   (1.2) power to thrusters via ESC
2. Deploy the USV on water.
3. Open the smartphone application.
4. Check sensor data and GPS data.
5. Control the USV through tapping buttons displayed on the smartphone screen.

7. Validation and characterization

Tests in an indoor swimming pool were used to verify the mobility and stability performance of the platform when given commands via a smartphone application (Fig. 9). We additionally performed field tests in a variety of outdoor, real-world environments to validate the usability of the platform. Those environments included a retention pond at Purdue Research Park, West Lafayette, IN, USA (Fig. 10a). In this experiment, the average temperature, turbidity, and pH measured with the on-board sensors were 6.190 °C, 2180.87 NPU, and 8.557, respectively.

In addition, we tested the platform at the Tuti intake station in Peru (Fig. 10b). The Tuti intake is at an altitude of 3,837 meters above sea level and has an average flow of 18 m^3/s. It has a dam that collects water from melting glaciers for use in irrigation of the Pampas de Majes, and is one of the most important irrigation projects in Peru [24,25]. The average temperature, turbidity, and pH measured with the on-board sensors during this experiment were 13.559 °C, 2380.89 NPU, and



(a) (b) (c)

Fig. 8. Android applications: (a) initial page (b) setup page, and (c) USV control and sensor reading page.

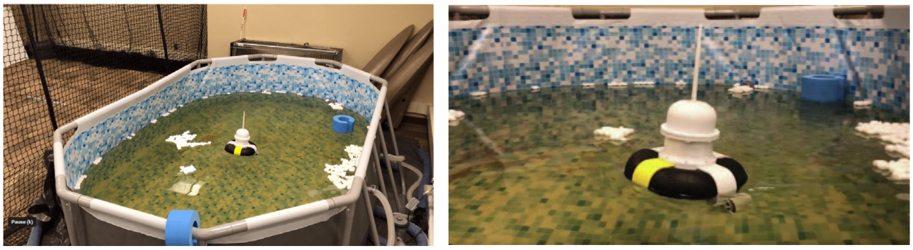


Fig. 9. Experiments of SMARTBoat 3 in an indoor swimming pool.

(a) (b) (c)

Fig. 10. SMARTBoat 3 field tests in (a) West Lafayette, IN, USA, (b) Tuti, Peru, and (c) Majes, Arequipa, Peru.

9.199, respectively. We also tested our platform in one of the irrigation retention ponds in Majes, Peru (Fig. 10c). The average temperature, turbidity, and pH in this pond were 27.842 °C, 2656.93 NPU, and 8.780, respectively. Thus, these experiments successfully validated the overall performance of the platform including mobility, stability, and usability. All test videos are available at the video links below:

- **Indoor experiment videos (indoor swimming pool)**
  -https://youtu.be/vqECeVWu_DI
  -https://youtu.be/IujafG1nARE
- **Outdoor experiment video at Purdue Research Park, West Lafayette, IN, USA**
  -https://youtu.be/-qeTTkvxu-M
- **Outdoor experiment video at Tuti, Peru**
  -https://youtu.be/nCS50jPCurQ

8. Application

This section presents a potential application and approach for enabling the SMARTBoat 3 to have autonomous capability by utilizing Bluetooth protocols and adding a new software application (such as the Robot Operating System (ROS) [26]). As it is shown in Fig. 11, the system consists of four nodes: (1) Nmea_navast_driver node: generates a NavsatFix ROS message

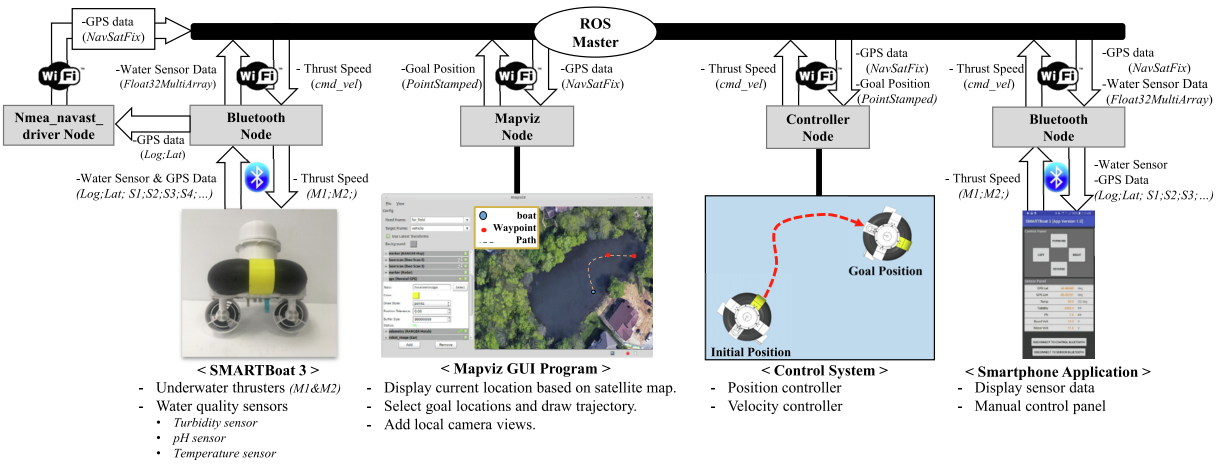


Fig. 11. Example of an autonomous program using the ROS system.

from GPS data; (2) Bluetooth node: transform wireless data into serial data or vice versa for the ROS master and other nodes; (3) Mapviz node: a visualization tool for visualizing camera views, GPS data, trajectory, and goal positions on satellite maps [27]; and (4) Controller node: calculates the speed needed from each thruster to get to the goal position.

Each node in Fig. 11 is able to communicate with the others and to exchange data via TCP/IP communication with 10 Hz sampling time; the maximum communication range is about 92 meters, depending on environmental conditions. Therefore, an autonomous system (or navigation) requires additional components, consisting of a WiFi router and a desktop/laptop. The detailed procedure for establishing autonomous navigation is listed below:

1. Install the WiFi router on the SMARTBoat 3 to enable ROS communication (TCP/IP)
2. Install *Ubuntu 16.04* and *ROS Kinect* on a computer (for outdoor experiments, a laptop is recommended rather than a desktop):
   - Ubuntu 16.04: https://ubuntu.com/
   - ROS Kinect: http://wiki.ros.org/kinect
3. Download *Mapviz* and *nmea_navsat_driver* packages from the ROS repository:
   - Mapviz node: http://wiki.ros.org/mapviz
   - nmea_navsat_driver: http://wiki.ros.org/nmea_navsat_driver
4. Program the Bluetooth nodes to convert Bluetooth data into ROS standard messages [28] and vice versa.
   (a) Before editing the code, connect two HC-06 Bluetooth modules and the controlling smartphone to the computer.
   (b) Search for each Bluetooth address; HC-06 addresses will begin with "98:D3:37".
   (c) (optional) Download the *PyBluez* Python library to enable easy access to the Bluetooth devices:https://github.com/pybluez/pybluez
   (d) Use ROS to edit the programming of the Bluetooth and controller nodes.
5. Publish all nodes/messages to ROS Master.
6. Display the current location and trajectory of the boat on the satellite map provided by the Mapviz GUI.
7. Make a controller system (such as a position/velocity controller), and then connect it with the ROS master.
8. Generate single/multiple goal positions using the Mapviz GUI program, and publish the position data to the ROS master.
9. Subscribe to the goal position data in the controller program, and then publish calculated speeds for each of the thrusters.
10. Subscribe to the thruster speed messages in the Bluetooth nodes, then convert them to Bluetooth-appropriate format for sending to the MCU to control the thrusters.

9. Conclusion

In this paper, we presented a new, fully open-source, small-sized and low-cost USV platform that can be used for near-surface water quality monitoring in real-time. Most parts of the platform (more than 90%) are printed using a 3D printer, and all hardware CAD design files, software source code, and the controlling Android application are publicly available, enabling people to easily assemble and modify the platform design and programming. We tested the platform through the indoor swimming pool and the field experiments in Peru and USA, successfully validating the performance of the platform in terms of usability, mobility, and stability. Additionally, we provided a potential application and approach with detailed procedures for enabling autonomous navigation using ROS middleware.

The proposed platform has great potential in the water environment monitoring field. For example, it includes a GPS module that allows the use of autonomous navigation technology, enabling automatic and regular water quality monitoring at multiple locations. Its communication functions allow water quality data measured by the USV to be transmitted to a web server, and water quality information to be shared with the public through a website. In addition, the inexpensive nature of this platform allows the manufacture of multiple units for simultaneously monitoring water quality with multiple USVs and in multiple locations. This will improve overall performance through shorter operating times, increased efficiency, and improved productivity compared to deployment of a single USV. Furthermore, the proposed platform can be an efficient and interesting educational tool for students to study environment and ocean engineering fields.

Declaration of Competing Interest

None.

Acknowledgements

This work was supported in part by the Purdue Research Foundation Graduate Fellowship and by the Arequipa Nexus Institute. The authors also thank the many members of the research team who assisted with different aspects of this work; notably Shyam Sundar Kannan, Jee Hwan Park, and Tamzidul Mina.

References

[1] M. Ertsen, T. Swiech, C. Machicao Pererya, "Reservoir storage and irrigation in arequipa, Peru," in EGU General Assembly Conference, Abstracts 12 (2010) 5579.
[2] A.M. Michalak, E.J. Anderson, D. Beletsky, S. Boland, N.S. Bosch, T.B. Bridgeman, J.D. Chaffin, K. Cho, R. Confesor, I. Daloğlu, Proc. Nat. Acad. Sci. (2013) 6448–6452.
[3] B. Qin, G. Zhu, G. Gao, Y. Zhang, W. Li, H.W. Paerl, W.W. Carmichael, A drinking water crisis in lake taihu, china: linkage to climatic variability and lake management, Environ. Manage. 45 (1) (2010) 105–112.
[4] M.V. Storey, B. Van der Gaag, B.P. Burns, Advances in on-line drinking water quality monitoring and early warning systems, Water Res. 45 (2) (2011) 741–747.
[5] J.B. De Sousa, G.A. Gonçalves, Unmanned vehicles for environmental data collection, Clean Technol. Environ. Policy 13 (2) (2011) 369–380.
[6] D.F. Carlson, A. Fürsterling, L. Vesterled, M. Skovby, S.S. Pedersen, C. Melvad, and S. Rysgaard, "An affordable and portable autonomous surface vehicle with obstacle avoidance for coastal ocean monitoring," Hardwarex, p. e00059, 2019.
[7] J.E. Manley, Unmanned maritime vehicles, 20 years of commercial and technical evolution, in: OCEANS 2016 MTS/IEEE Monterey, Sep. 2016, pp. 1–6.
[8] C. Scholin, G. Doucette, S. Jensen, B. Roman, D. Pargett, R. Marin III, C. Preston, W. Jones, J. Feldman, C. Everlove, et al, Remote detection of marine microbes, small invertebrates, harmful algae, and biotoxins using the environmental sample processor (esp), Oceanography 22 (2) (2009) 158–167.
[9] P. Johnston, M. Poole, Marine surveillance capabilities of the autonaut wave-propelled unmanned surface vessel (usv), OCEANS 2017-Aberdeen, IEEE, 2017, pp. 1–46.
[10] Y. Wang, N. Huang, Y. Yang, Platypus cooperate robotic watercraft platform, Summer Scholar Program (2014) 78.
[11] A. Tinka, M. Rafiee, A.M. Bayen, Floating sensor networks for river studies, IEEE Syst. J. 7 (1) (2013) 36–49.
[12] Y. Kaizu, M. Iio, H. Yamada, N. Noguchi, Development of unmanned airboat for water-quality mapping, Biosyst. Eng. 109 (4) (2011) 338–347.
[13] J.A.L. Castilla, "Jalc boat, aquatic robot platform." [Online]. Available: https://www.instructables.com/id/JALC-Boat-Aquatic-Robot-Platform/.
[14] "HC06 Serial Bluetooth Brick – ITEAD Wiki." [Online]. Available: https://www.itead.cc/wiki/HC06_Serial_Bluetooth_Brick.
[15] T. TheNavalArch, Bollard pull calculations - an introduction (part i), May 2019. [Online]. Available: https://thenavalarch.com/bollard-pull-calculations-introduction/.
[16] Dangerous waves and your boat. [Online]. Available: http://www.oceannavigator.com/Ocean-Voyager-2011/Dangerous-waves-and-your-boat/..
[17] S. Kirby, tgy – open source firmware for atmega-based brushless escs. [Online]. Available: https://github.com/sim-/tgy.
[18] Lulzbot taz 6. [Online]. Available: https://www.lulzbot.com/store/printers/lulzbot-taz-6.
[19] eSUN PLA Filament. [Online]. Available: http://www.esun3d.net/.
[20] Rov and marine robotics systems and components. [Online]. Available: https://bluerobotics.com/.
[21] S. Yilmaz, K. Bayramoglu, Remotely operated vehicle (rov) design and fuel cell applicability, World Electro Mobil Conference, 2017, pp. 114–118.
[22] E.H. Binugroho, R.S. Dewanto, D. Pramadihanto, erov: Preliminary design of 5 dof rov using 6 thrusters configuration, International Electronics Symposium on Engineering Technology and Applications (IES-ETA), IEEE, 2018, pp. 281–287.
[23] Quality arduino robot iot diy electronic kit. [Online]. Available: https://www.dfrobot.com/.
[24] J.O. Maos, Water resource development and land settlement in southern Peru: The Majes Project, GeoJournal 11 (1) (1985) 69–78.
[25] "Majes siguas special project." [Online]. Available: http://www.bcrp.gob.pe/docs/Proyeccion-Institucional/Encuentros-Regionales/2016/arequipa/eer-arequipa-2016-ramos.pdf.
[26] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A.Y. Ng, "Ros: an open-source robot operating system," in ICRA workshop on open source software, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.
[27] "Mapviz - ROS Wiki." [Online]. Available: http://wiki.ros.org/mapviz.
[28] "std_msgs - ROS Wiki." [Online]. Available: http://wiki.ros.org/std_msgs.