

Chapter 8

Low-Cost Unmanned Surface Vehicle for Autonomous Bathymetric Surveillance



Vladimir Nikishin, Maxim Durmanov, and Ivan Skorik

8.1 Introduction

The main goal of current development is to design complex solution including unmanned surface vehicle (USV) and custom autopilot firmware and software for real-time depth visualization. In the beginning of development, we have pointed next requirements for the USV:

- Easily accessible low-cost hardware parts
- Easy USV transportation
- Monohull
- Free software libraries and IDE
- Custom autopilot
- Custom base ground station
- Propulsion high reliability
- Maneuverability
- Minimal continuous USV work over 1 h
- Custom software for collected data post-processing

For suitable USV transferring, we have determined the longest dimension is 1 meter, which is dependent on average size of car trunk. Monohull shape was chosen because of simple manufacturing and because USV oriented to work in calm waters such as lakes and internal bays where the high waves are rare.

V. Nikishin · M. Durmanov · I. Skorik (✉)
Sevastopol State University, Sevastopol, Russia
e-mail: IVSkorik@sevsu.ru

© The Author(s), under exclusive license to Springer Nature
Switzerland AG 2021

S. Bauk, S. D. Ilčev (eds.), *The 1st International Conference on Maritime Education and Development*, https://doi.org/10.1007/978-3-030-64088-0_8

8.2 USV Main Parts

Hull

USV was made of two main materials – fiberglass and styrofoam as hull base. The hull base was CNC-machined and glued together in one part (Fig. 8.1).

Then styrofoam was covered by three layers of fiberglass (Fig. 8.2).

For minimizing interaction of electronic parts with water, they are placed over the top in separated covered box mounted with PVC pipes glued to the USV hull. Placing antennas as high as possible also gives advantages for reliable connection with ground station and easy access to electronic; it worthwhile because of magnetometer that is far enough from engine containing strong magnets. Final results of the hull are shown in Fig. 8.3.

Engine and Propulsion System

We have selected electrical propulsion system because modern brushless and brushed motors have enough power and they have small size suitable for use in USVs. Electrical propulsion system allows fast changing battery, and it does not pollute water by any hazardous materials or [exhaust gases](#). Experimental USV uses 3S (12 volts) battery and 4074-sized brushless motor with 2000 kV and total power over 1 kW. Motor is driven by electronic speed controller (ESC). Because ESC dissipates a lot of heat, it requires cooling; cooling is produced by water injected by water jet.

For steering and movement of USV, we used water jet. Water jet has advantages in safety and maneuverability. Safety reason is very important if USV works near the beaches with swimming people. Because water jet rotor is located inside tube, it is almost impossible to injure someone by rotating blades. Open propellers are more dangerous; also, there are big chances that floating garbage, seaweed, or fish lines could wound up on the shaft and propeller blades. Even if water jet has no safety lattice, there are always problem of floating garbage and seaweed (Fig. 8.4).

Fig. 8.1 CNC-machined styrofoam hull base

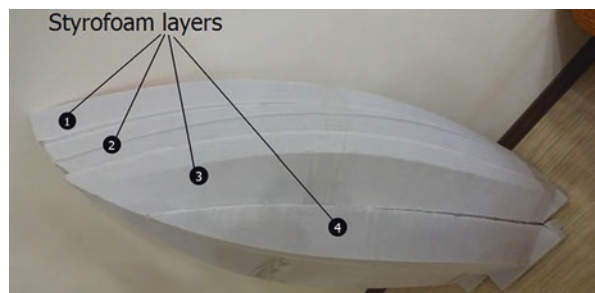


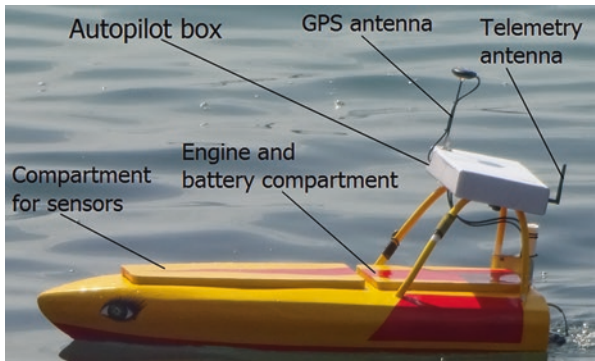
Fig. 8.2 Hull covering by fiberglass



Fig. 8.4 Seaweed in water jet



Fig. 8.3 Unmanned surface vehicle



For steering of USV was used rotating nozzle. Nozzle has rotating angle near 30° left and right. This allows to get very good maneuverability.

Autopilot Hardware

The main purpose of autopilot is to control USV by rotating nozzle to steer it at given route point. This requires sensors (gyro, accelerometer, and magnetometer) integration with microcontroller (MCU). We selected Atmega2560 on Arduino Mega board [1]. This MCU has enough interfaces for connecting with sensors and does not require high programming skill. For autonomous movement of USV, main measurements are current heading and current location. Heading is measured by MEMS magnetometer connected to MCU by I2C interface. Magnetometer requires calibration to minimize hard and soft iron distortion. For measurement, location of USV uses GPS module U-Blox connected to MCU by UART interface. For controlling nozzle and motor, ESC uses PWM signal generated by internal hardware timer. We have designed PCB sandwich board inserted in MCU board (Fig. 8.5).

Autopilot Software

Firmware was built using free Atmel Studio IDE. Firmware includes functions for azimuth and distance calculation to given point and converting controlling commands to PWM signal for servo and water jet nozzle.

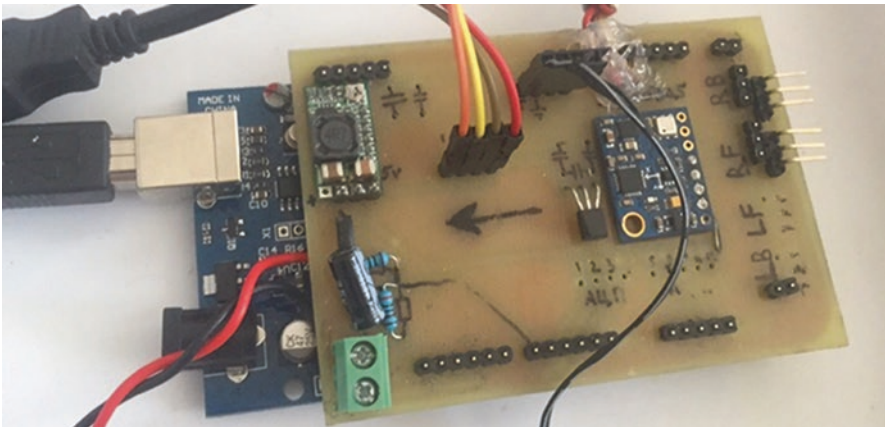


Fig. 8.5 Autopilot

Sonar

The most useful part of USV development is integration with measurement devices. Current development is targeted to bathymetry. Selection of sonar depends on many reasons as precision, resolution, and single- or multi-beam ray. However, in the case of USV, the most important is connection with MCU – sonar has to be equipped with suitable interface; in most cases this is RS-232 with NMEA standard messages [2]. We choose Garmin Echomap 42 [3]; this sonar has 5 volts NMEA interface, but its signal is inverted. Because Atmega2560 does not have an option to invert UART signal, we have to add an inverter before signal goes to MCU using NPN transistor (Fig. 8.6).

Placing sonar in internal compartment protects it from water sprays (Fig. 8.7).

Fig. 8.6 NMEA inverter

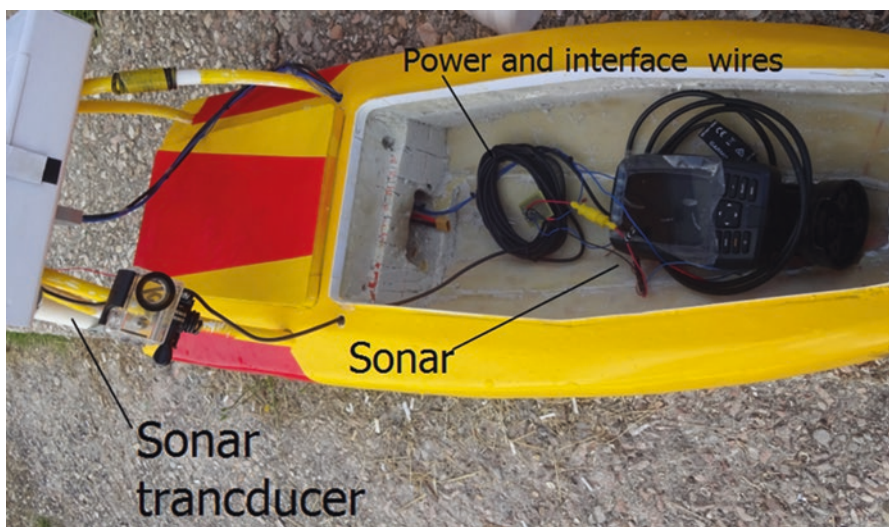
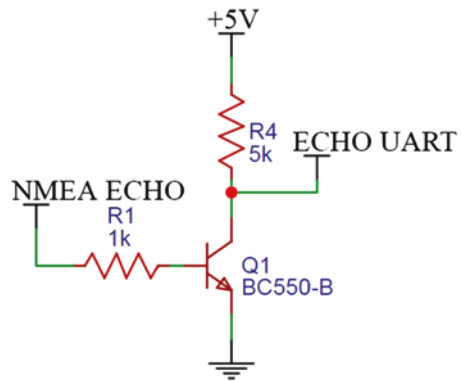


Fig. 8.7 Placing sonar in USV

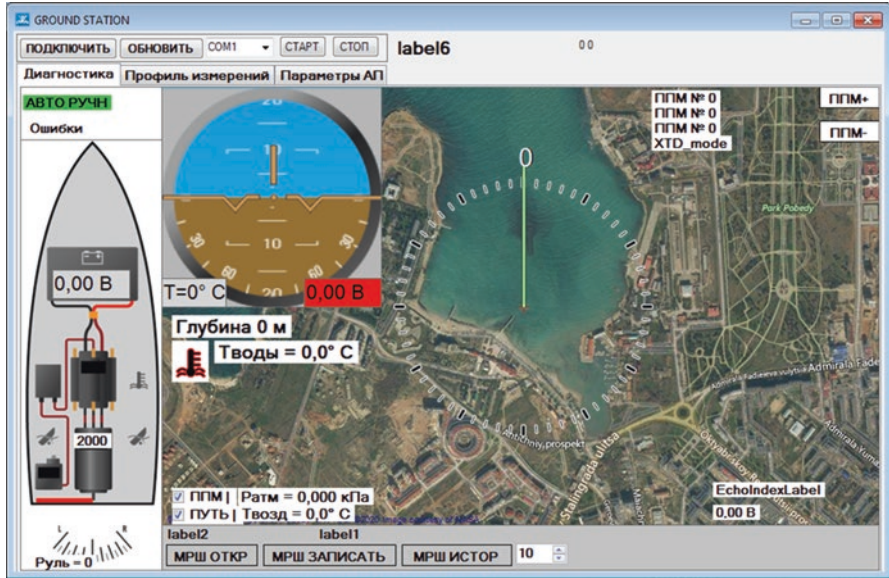


Fig. 8.8 Ground station GUI

Ground Station Software

Ground station (GS) visualizes USV data in real time. Graphical user interface (GUI) of GS is shown in Fig. 8.8. GS was built in free Microsoft Visual Studio C# using free GMap.Net library for map positioning of USV. GS receive data from USV using wireless 915 MHz telemetry modules. The received telemetry stores on PC in text document file for further post-processing. Post-processing could be done with software as Golden Surfer or ReefMaster, but we have designed our custom software.

8.3 Post-processing Software Development

Statement of the Bathymetric Data Visualization Problem

It is proposed to visualize bathymetric data using a heat map. To do this, select a color palette, and fill with color each area with the same depth. The initial data for building a heat map are:

- Geographical coordinates of points (latitude and longitude) and depth values measured by the sonar at these points

- Geographical coordinates of the boundary points of the test site, to create the boundaries of the displayed heat map
- Color palette (range and number of color shades)

To solve this problem, there are special services. For example, the HeatmapTool [4] online service allows you to use various color palettes; control the radius, zoom, and opacity of heat spots; and update the map in real time. However, this service is paid and designed to display statistical data in the selected region.

Heat Map Algorithm

The problem of building a heat map is not only to fill on the map the individual points at which depth measurements were made but to fill the entire area with the color. This problem is solved using the inverse distance weighting (IDW) [5] method, which consists in determining the weight of each point by interpolating neighboring points with a known weight at a given point.

The heat map algorithm consists of the following steps:

1. Converting the geographical coordinates of the points to the x and y coordinates of the Cartesian coordinates
2. Formation of the border of the polygon for cropping the resulting image of the heat map
3. Implementation of the IDW method for constructing a heat map image
4. Converting points in Cartesian coordinates to geographical coordinates and combining a heat map with a geographical map of the area

Geographic coordinates and depth measurements are taken from the sonar and saved in a text file. The geographic coordinates of the boundary points are taken in advance from the terrain map and saved in a file. The color palette is chosen arbitrarily; usually warmer tones correspond to smaller depths and colder ones to larger ones. The algorithm is implemented using developed software. An example of building a depth map for an artificial reservoir in Inkerman (Sevastopol) is shown in Fig. 8.9 and the coastal waters of Holland Bay (Sevastopol) is shown in Fig. 8.10.

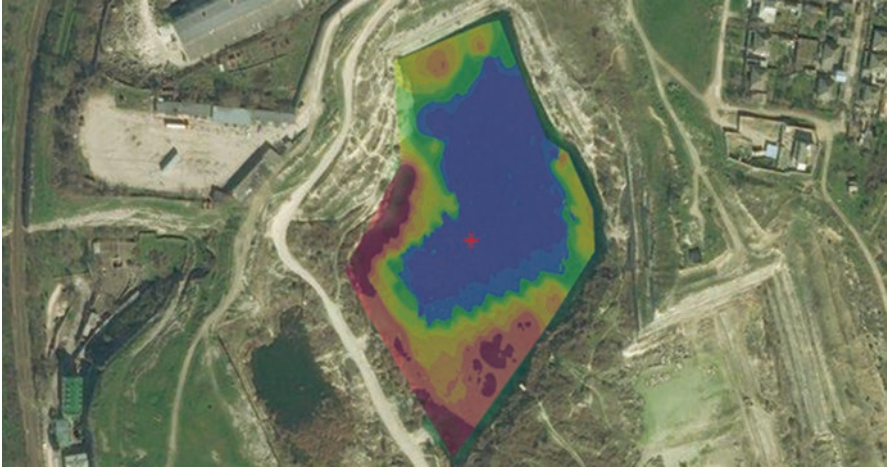


Fig. 8.9 Heat map of Inkerman (Sevastopol)

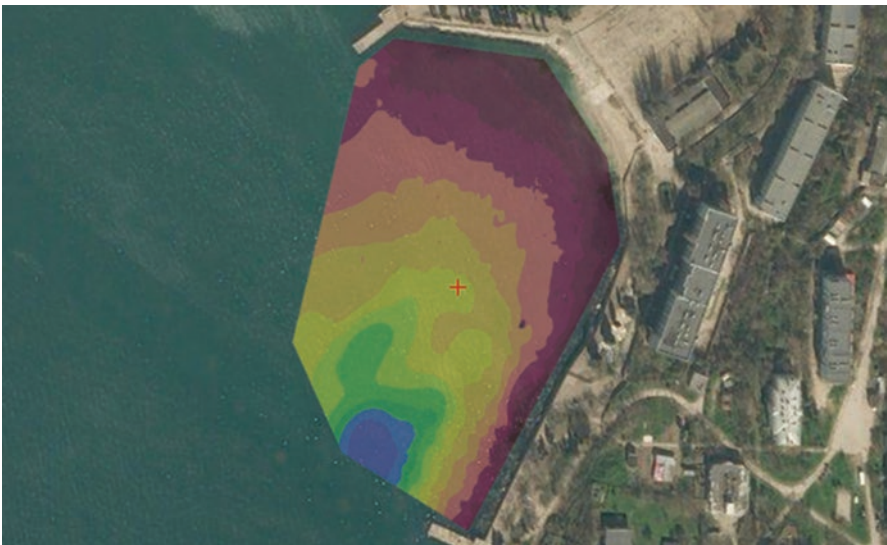


Fig. 8.10 Heat map of Holland Bay (Sevastopol)

8.4 Conclusion

Developed USV shown in the paper has simple design which can be reproduced with minimal cost. In summary, the price of bill of materials without sonar is less than 500\$. The main limitation of our USV is it works only in calm waters without strong wind and waves. Small draft of USV less than 3 inches allows mapping the depth of lakes or bays very close to shoreline; this is almost impossible with big boats with crew on board.

Acknowledgment The reported study was funded by the internal grant of Sevastopol State University, project number 516/06-31.

References

1. Arduino Mega official site, <https://store.arduino.cc/arduino-mega-2560-rev3>
2. National Marine Electronics Association, <https://www.nmea.org>
3. Garmin Echomap sonar, <https://buy.garmin.com/en-US/US/p/592785>
4. HeatmapTool.com, <https://www.heatmaptool.com>
5. GisGeography. Inverse Distance Weighting (IDW) Interpolation, <https://gisgeography.com/inverse-distance-weighting-idw-interpolation>