M. Chen et al. (Eds.)

doi:10.3233/ATDE230504

Design of a Control System for an Autonomous Underwater Vehicle EDYSYS1

Edosa OSA [a,1], Daniel Chinemelem SAMUELS [b]
[a] *Department of Electrical and Electronic Engineering, University of Benin, P.M.B. 1154, Benin City, Nigeria*
[b] *Department of Structural Engineering, University of Benin, P.M.B. 1154, Benin City, Nigeria*

Abstract. This paper describes the design of a control system for an autonomous underwater vehicle EDYSYS 1. An autonomous underwater vehicle is an unmanned and self-propelled underwater vessel that can operate independently underwater and carry out several missions as assigned to it compared to a Remotely Operated Vehicle (ROV) which is usually tethered to a ship or some other moored water vehicle. Intelligent design of control systems for autonomous underwater vehicles is an active area of research giving the demands for autonomy and the capacity for intelligent systems to satisfy such demands. A control system has been designed with the Raspberry Pi 4 computer as the main control unit. Various subsystems and sensors for data acquisition by the vehicle are controlled by the Raspberry Pi 4 which has the Robot Operating System (ROS) configured in it. The necessary intelligence for controlling each sensor is configured using Python programming language. The relevant Python scripts are thereafter implemented as nodes within the ROS framework. By calling the relevant nodes in ROS, various values of sensory data were obtained by the designed system in the ROS environment. Successful communication via LoRa was also achieved.

Keywords. Autonomous, raspberry Pi, sensor, ROS

1. Introduction

Autonomous underwater vehicles (AUVs) are unmanned, self-propelled vehicles that are typically deployed from a surface vessel and can operate independently of that vessel for periods of few hours to several days [1]. Recently, researchers have focused on the development of AUVs for long-term data collection in oceanography and coastal management [2]. Scientists use AUVs to study lakes, the ocean body as well as the ocean floor. A variety of sensors can be affixed to AUVs to measure the concentration of various elements or compounds, underwater temperature, pressure, the absorption or reflection of light, presence of microscopic life, etc. Research into development of low cost autonomous underwater vehicles is an active domain. Also, the autonomy of underwater vehicles is a major issue which can be improved by using intelligent systems up to certain extent [3]. With the rapid developments of both computing

[1] Edosa OSA, Corresponding author, Department of Electrical and Electronic Engineering, University of Benin, P.M.B. 1154, Benin City, Nigeria; E-mail: edosa.osa@uniben.edu.

hardware and artificial intelligence in recent years, such implementations achieve encouraging performance particularly in the condition of making plans and decisions for complex systems in which it is difficult to establish a set of accurate models such as underwater. By intelligent acquisition of environmental data, the AUV can make decisions as touching control and navigation.

Authors in [4] designed a wireless mobile underwater and surface vehicle. It involved the use of low-cost material like commercial grade polyvinyl chloride (PVC) pipes. However, the vehicle utilized solar panels to charge the batteries which places a limit on the vehicle size. Authors in [5] developed the Vidyut AUV that carries out underwater obstacle avoidance and object recognition using appropriate devices and sensors. The main control intelligence was implemented via an Arduino Uno microcontroller. However, in terms of speed, the Arduino Uno performs less than the Raspberry Pi and its Portenta H7 board has less CPU power than the Raspberry Pi 4 computer. Furthermore, the Pi computer has more functionality and access to many different programming languages since it runs on Linux [6]. The EDYSYS 1 AUV (shown in figure 1) whose control system design is described in this paper was implemented using the Raspberry Pi 4 computer running on the Raspbian Linux operating system.

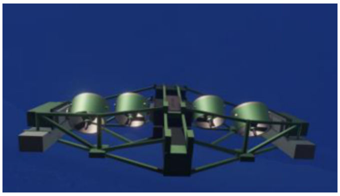

Figure 1. CAD Design of EDYSYS 1 Vehicle in Blender Environment.

The rest of the paper is organized as follows: Section 2 describes the Control system Design. Section 3 presents the Experimental Results. Section 4 presents the Discussion of Results. Finally, Section 5 presents the Conclusion.

2. Control System Design

The design of the control system is modular with each module connected to a control unit, namely a Raspberry Pi 4 computer. Figure 2 shows the block diagram of the control system. The subsystems are described thereafter.

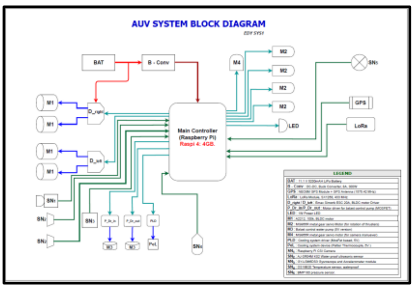

Figure 2. Control System Block Diagram.

2.1. Main Control Unit

The EDYSYS 1 AUV is controlled by a Raspberry Pi 4 computer as shown in figure 3. The Robot Operating System (ROS) Package was configured into the Pi along with relevant python scripts for the AUV sensors.

This unit has the following specifications [7]:

- Processor: 1.5 GHz 64-bit Quad Core ARM Cortex-A72
- GPU: Video Core IV Graphics
- GPIO: 40 Pins
- RAM: 4GB
- I/O Interfaces: 2x 2.0 USB Ports, 2x 3.0 USB Ports, 1 Full HDMI port, Camera Interface
- Power Source: Micro USB or GPIO
- SD Card Slot: Micro Size SD Card Slot

2.2. Power Supply Input

It consists of two 11.1V 5200mAH LiPo batteries and one DC-DC Buck converter 8A, 300W, switches and voltage regulators. The 12V IN is from the LiPo battery connection and it powers all components requiring 12V for their operation such as Thruster Relay. The Buck converter accepts 12V and gives out 5V (as shown in figure 4). The raw 5V and GND from the buck converter supply any component requiring 5V. C5 is a decoupling capacitor (100 microfarads) which serves to provide a local instantaneous charge source that prevents the voltage source from dipping and a bypass that dampens ringing due to the operation of inductive loads in the system (such as the Brushless DC motor and Servo motor). D3 is a polarity protection diode which protects against wrong connection of positive to negative.

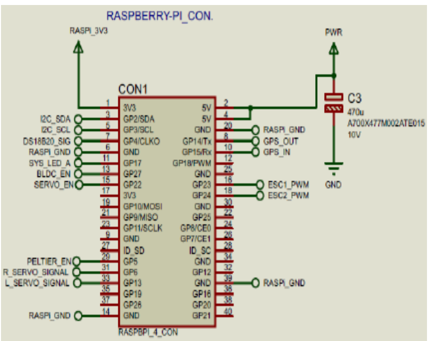

Figure 3. Raspberry Pi 4 Control Connection.

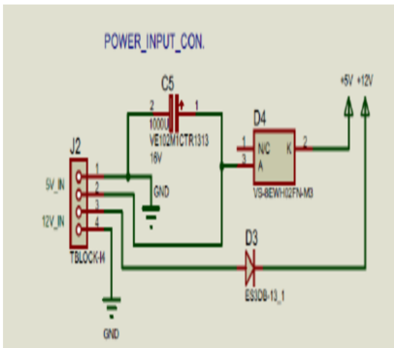

Figure 4. Power Supply Connection.

2.3. Electronic Speed Control System

This subsystem consists of four Electronic Speed Controllers (ESC) 12 Volts DC, four Brushless DC motors (BLDC), four Servo motors and a BLDC enable Relay as shown in figure 5. U4 is the ESC_SIGNAL_LEVEL_SHIFTER. Pulse Width Modulation (PWM) signal is required to control the speed of the BLDC and the position of the servos. However, the logic level of the Pi is 3.3V but that of the ESC is 5V, hence a level shifter is needed to transform the logic level of the Pi to 5V for optimum result. The ESC (Electronic Speed Controller) module interfaces the Raspberry Pi computer with the BLDC motors.

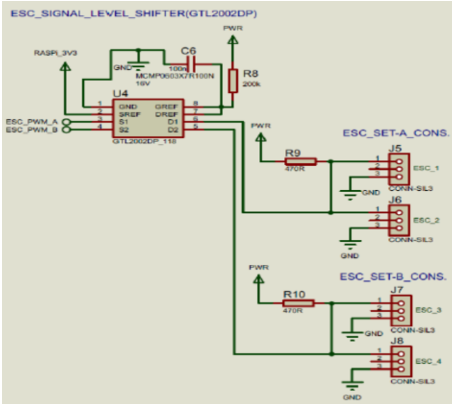

Figure 5. Electronic Speed Control Connection.

2.4. Data Acquisition Subsystem

This consists of the following sensor modules:

Pressure Sensor: This sensor is responsible for sending information about depth and pressure. I^2C (Inter Integrated Circuit) connection is used to integrate this sensor to the Pi.

Laser Sensing Unit: This unit is used to sense the distance of the AUV from an impending obstacle within its frontal axis. It consists of two laser ranging sensors, one at the right and the other at the left of the AUV. It also operates via I^2C and is powered via the 3.3V regulator.

Global Positioning Satellite (GPS) Module: The GPS module also receives its supply from the 3.3V regulator. It uses Universal Asynchronous Receiver/Transmitter (UART) to communicate with the Pi. However, in this work, only the transmit pin of the GPS module is connected to the receive pin of the Pi since it will only be sending GPS coordinates via serial protocol to the Pi.

Gyroscope/Accelerometer: This sensor is used for obtaining the angular orientation signals of the AUV. It is also integrated to the Pi through I^2C with a unique slave address.

Temperature Sensing Unit: The bmp180 sensor is used along with ds18b20 sensor for obtaining temperature reading of AUV environment.

2.5. Cooling Subsystem

This unit consists of a Peltier device and the Raspberry Pi Heat Sink. The Peltier device functions to dissipate the heat generated in the enclosed AUV, while the Pi heat sink dissipates heat specifically from the Pi. The Peltier device receives a PELTIER_ENABLE signal from the Pi for it to function.

2.6. Communication Subsystem

This consists of a twin pair of Ashining AS32 LoRa module that served as a medium of communication between the Raspberry Pi and a laptop computer. The frequency of operation is 433MHz.

3. Experimental Results and Discussion

The various python scripts representing the sensor nodes of the AUV were executed in the ROS environment and results were obtained at different instances as shown in figures 6 to 10.

Figure 6. Data Acquisition with bmp 180 Node

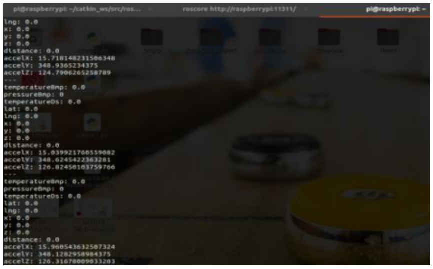

Figure 7. Data Acquisition with gyro Node

Figure 8. Data Acquisition with joystick Node

Figure 9. Data Transmission via lora.py Script

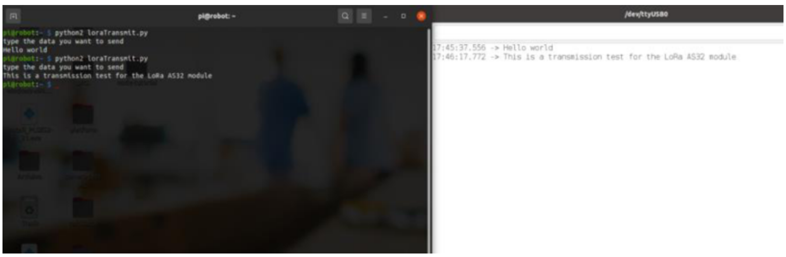

Figure 10. Communication via LoRa Modules.

Figures 6 to 8 describe various readings obtained by the AUV peripherals. Figures 9 and 10 describe the communication process performed by the LoRa modules. These readings for sensor values were obtained by calling the programmed python scripts for the respective nodes in the configured Robot Operating System (ROS) within the Raspberry Pi 4 controller. By implementing ROS, the bmp 180 sensor Node could obtain readings for temperature and pressure as displayed in Figure 6. Similarly, the gyro could obtain the acceleration in the x, y and z directions (representing right-left turn, forward turn and speed respectively) as shown in Figure 7, while the joystick node obtained the x, y and z orientation values for the designed AUV as shown in Figure 8. Communication was also established between the LoRa modules as described in Figures 9 and 10.

4. Conclusion

This paper describes the design of the control system of an autonomous underwater vehicle. The Raspberry Pi 4 served as the main control unit for the vehicle. Various sensors were attached to the AUV for data acquisition purposes and Robot Operating System (ROS) framework was implemented in the Pi computer. Calling the relevant nodes in ROS resulted in the operation of the AUV sensors to obtain readings as shown.

References

[1] Wynn R.B, Huvenne V.A.I, Le Bas T.P, et al. Autonomous Underwater Vehicles (AUVs): Their past, present and future contributions to the advancement of marine geoscience. Marine Geology. 2014; 352: 451-468.

[2] Saghafi M and Lavimi R. Optimal design of nose and tail of an autonomous underwater vehicle hull to reduce drag force using numerical simulation. Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment. 2020; 234 (1): 76–88. doi:10.1177/1475090219863191. ISSN 1475 0902. S2CID 199578272.

[3] Sahu BK and Subudhi B. The State of art of autonomous underwater vehicles in current and future decades. 2014 First International Conference on Automation, Control, Energy and Systems (ACES), 2014, pp. 1-6, doi: 10.1109/ACES.2014.6808014.

[4] Okunlola BA, Okogbue EC, Ewetumo T, Olabanji OM and Adeboye AOC. Design, construction and control of a wireless mobile underwater and surface vehicle for hydro-meteorological parameters monitoring. International Journal of Engineering Research & Technology (IJERT), 2020; 9(10).

[5] Abishek SM, Shanmugasundar G, Mathesh, Sairam T and Rajkumar P. Design, analysis and development of autonomous underwater vehicle. International Journal of Innovative Technology and Exploring Engineering (IJITEE). 2019; 9(2).

[6] Pounder L. Raspberry Pi vs Arduino: Which Board is Best? Available at: www.tomshardware.com (Accessed Mar. 13 2023).

[7] Khan A. Raspberry Pi History' Available at: https://linuxhint.com/ Accessed 25 Mar. 2023.