EVOLUTION IN ELECTRICAL AND ELECTRONIC ENGINEERING
e-ISSN: 2756-8458

EEEE

Vol. 5 No. 1 (2024) 427-435
https://publisher.uthm.edu.my/periodicals/index.php/eeee

Design and Development of a Remotely Operated Underwater Vehicle (ROV)

Thineshwaran Satiamurti[1], Herdawatie Abdul Kadir[1]*

[1] *Department of Electronic Engineering, Faculty of Electrical and Electronic Engineering Universiti Tun Hussein Onn Malaysia, Batu Pahat, Johor, 86400, MALAYSIA*

*Corresponding Author: watie@uthm.edu.my
DOI: https://doi.org/10.30880/eeee.2024.05.01.054

Article Info

Received: 11 January 2024
Accepted: 15 March 2024
Available online: 30 April 2024

Keywords

Remotely operated Underwater Vehicle (ROV), NodeMCU ESP8266, IMU-MPU6050, Bar30 High-Resolution Depth sensor, ESP32-Cam, Blynk, ThingSpeak

Abstract

This work overcomes obstacles in traditional underwater exploration by creating a cutting-edge Remotely Operated Underwater Vehicle (ROV). Research in this field is captivating and holds immense potential. However, underwater activities challenges and hazardous conditions limit human involvement in these environments. Thus, there is a need for underwater exploration vehicles to the point humans cannot reach, including recording underwater data. The current underwater vehicles have notable constraints, limiting their capabilities and resulting in uninformed decision-making and costly errors due to the lack of real-time data transmission capabilities. The proposed ROV, featuring an open frame structure and microcontroller ESP8266, integrates advanced components such as the ESP32-Cam, IMU-MPU6050 sensor, depth sensor, and T200 Blue Robotic thrusters. The methodology includes comprehensive block diagrams detailing the overall remote system and the proposed design. The results demonstrate a finalized structure design, mechanical design arrangements, and ROV server connection. An optimized ROV structure with extensive functionality, connectivity, and reliability testing demonstrates ideal buoyancy, balance, and waterproofing capabilities. Incorporating a user-friendly Blynk app interface and seamless integration with ThingSpeak enhances real-time data monitoring. This innovative ROV work contributes to society by overcoming challenges in traditional underwater exploration.

1. Introduction

Underwater exploration holds immense potential, but challenges and hazardous conditions limit human involvement [1]. A dedicated tool, such as a Remotely Operated Underwater Vehicle (ROV), is needed to help humans survey and explore deep and complex areas [2]. ROVs can be classified into Manned Underwater Vehicles (MUV) and Unmanned Underwater Vehicles (UUV), with ROVs being controlled remotely by human operators [3].

The open space design of the ROV ensures its ability to withstand external pressure, improving its overall performance and functionality [4]. The existing methods used for underwater exploration and inspection pose significant challenges in terms of cost, time consumption, and the safety of human divers [5]. The current underwater vehicles available have notable constraints, limiting their capabilities and resulting in uninformed decision-making and costly errors due to the lack of real-time data transmission capabilities [6]. This research focuses on designing and developing a small ROV specifically tailored for underwater applications. The mechanical design and hardware aspects are thoroughly explained, along with the electrical components and sensors integrated into the ROV. The project aims to enhance the accessibility and usability of underwater exploration vehicles by designing a low-cost ROV and implementing a remote human-machine interface through the Blynk app. The inclusion of an ESP32 camera, IMU, and Depth sensors enhances the ROV's capabilities for accurate navigation and environmental monitoring.

2. Methodology

2.1 Project Development

Fig. 1 shows the block diagram for the overall remote system. In this project, the software used is the Blynk app, which sends directions and instructions to ROV using Wi-Fi. One aspect of the ROV's control system is the utilization of the Blynk app, which serves as the human-machine interface. The Blynk app allows users to control the ROV using a smartphone remotely. Through the app, the user can interact with virtual buttons, associated with specific functions or movements of the ROV. The thruster motion data will be sent to ThingSpeak cloud.

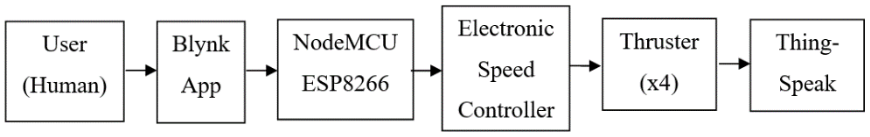


Fig. 1 *Block diagram of the remotely operated system*

Fig. 2 below shows the block diagram of the proposed design for this project. The prototype was realized using a depth sensor, IMU sensor, ESP32 camera, and NodeMCU Esp8266. Next, for the microcontroller will be used the NodeMCU ESP8266 is used as the microcontroller, and for the output, will see the result from the serial monitor at the Arduino IDE and mechanical out as thrusters and in IoT cloud data will be stored in ThingSpeak.

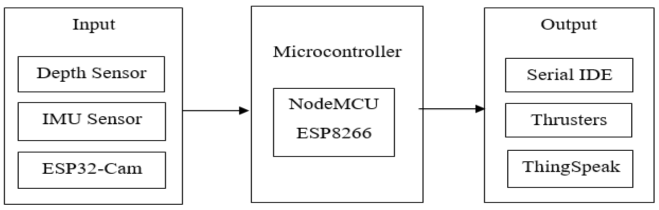


Fig. 2 *Block diagram for a proposed design*

2.2 ROV Design Process

The project involves designing a ROV, which consists of several phase steps. The first phase focuses on the electrical and mechanical nature of the main system as shown in Fig. 3. The next phase is divided into two parts: mechanical design and electrical design for both internal and external ROVs. SolidWorks is used for mechanical design, while Proteus software is used for electrical design.

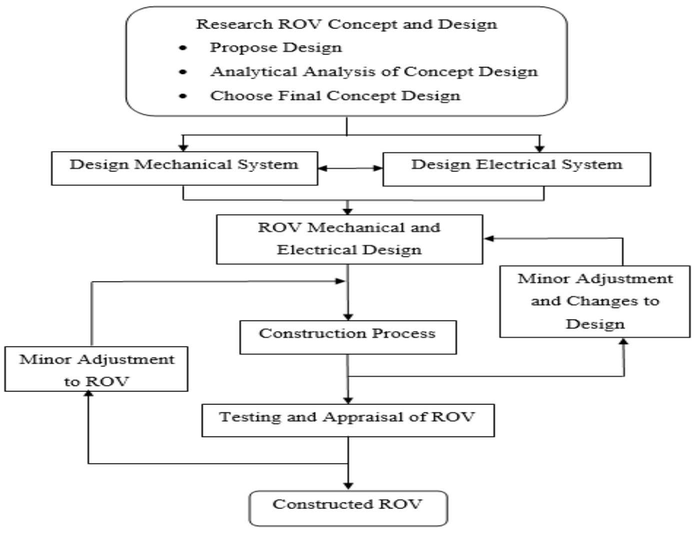
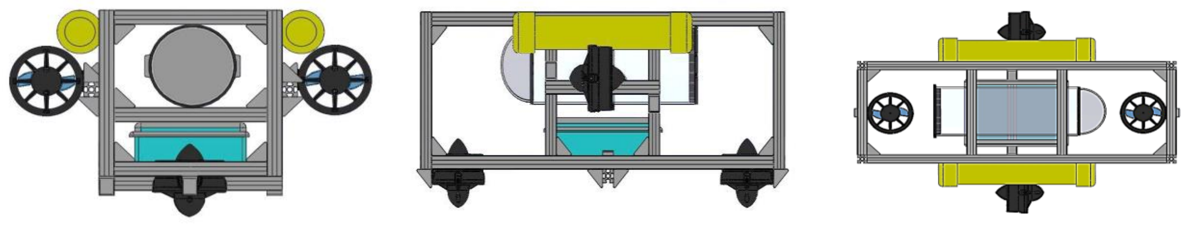

Research ROV Concept and Design
Propose Design
Analytical Analysis of Concept Design
Choose Final Concept Design
Design Mechanical System
Design Electrical System
ROV Mechanical and Electrical Design
Minor Adjustment and Changes to Design
Construction Process
Minor Adjustment to ROV
Testing and Appraisal of ROV
Constructed ROV

Fig. 3 *Block diagram for a proposed design*

2.3 ROV Mechanical and Electrical & Electronic Design Structure

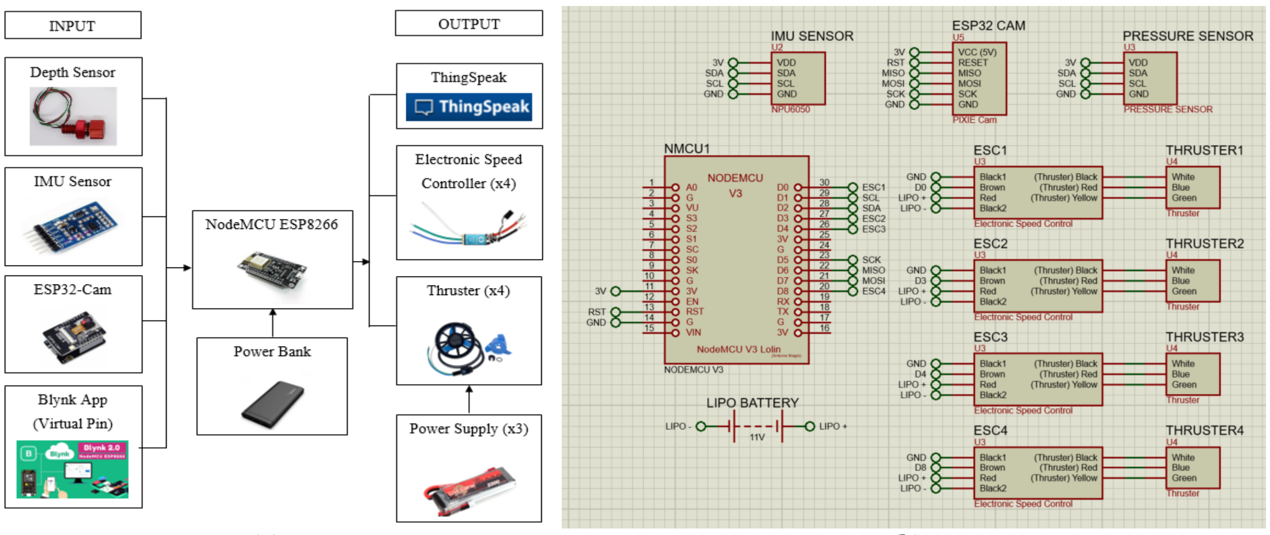
The SolidWorks software is used for ROV’s design structure. Fig. 4 shows the main design of the ROV accordingly to the designed dimension and parts that had been used. It shows the placement of all the ROV's mechanical parts and the main and primary compartments that contain all the ROV's electrical parts of the design.



(a) **(b)** **(c)**

Fig. 4 *ROV design view (a) Front; (b) Side; (c) Top*

For electrical and electronic modules, Fig. 5 shows the overall component used for the ROV, and the circuit design of a ROV drawn using Proteus Software. All electrical and electronic components used in this system are connected to a microcontroller NodeMCU Esp8266.



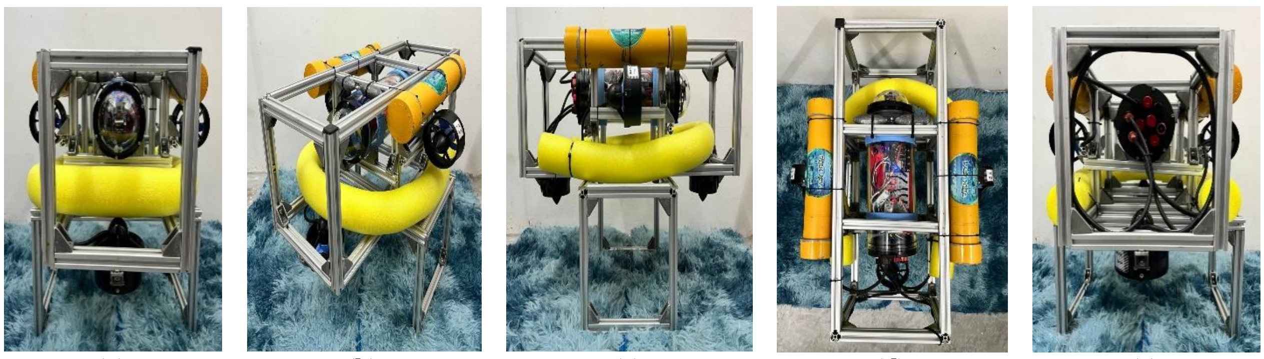
(a) **(b)**

Fig. 5 *Electrical & Electronic parts (a) E&E system; (b) Circuit design*

3. Results and Discussion

3.1 Finalized Structure Design

The finalized design of the ROV, as shown in Fig. 6, includes integrated mechanical and electrical components strategically positioned for optimal buoyancy and stability. The primary compartment houses the ESP8266 main circuit, IMU-MPU6050 sensor, Bar30 depth sensor, ESP32-Cam, power bank, and Li-Po battery. The thruster, foam hollow, motors, and sea sinker enhance the ROV's effectiveness outside the primary compartment, making its underwater operations more efficient.

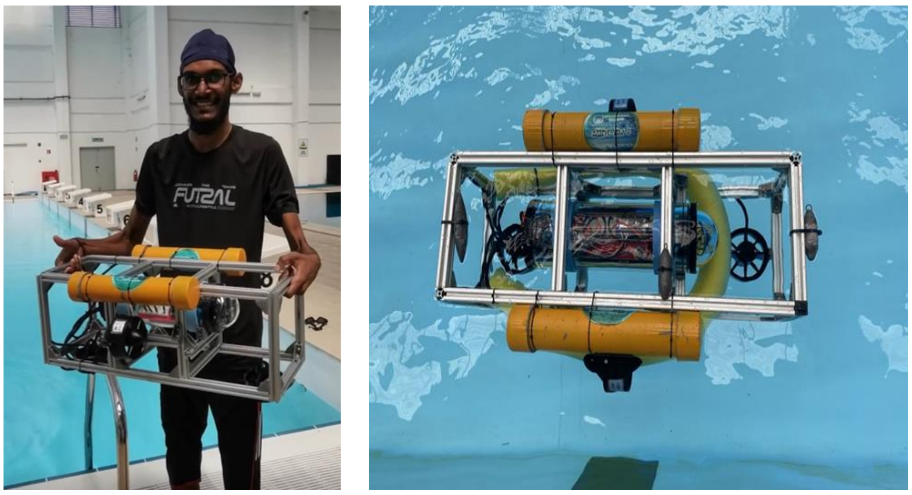


(a) **(b)** **(c)** **(d)** **(e)**

Fig. 6 *Finalized ROV view (a) Front; (b) Isometric; (c) Side; (d) Top; (e) Back*

3.2 ROV Buoyancy, Balance, and Waterproof Test

The ROV passes through comprehensive tests to ensure its buoyancy, balance, and waterproof capabilities. The primary compartment's electronic components are protected from water. Fig. 7 shows the before and after submerging the ROV and faces long-term submersion in static water in a swimming pool, identifying and fixing water leaks.



(a) (b)

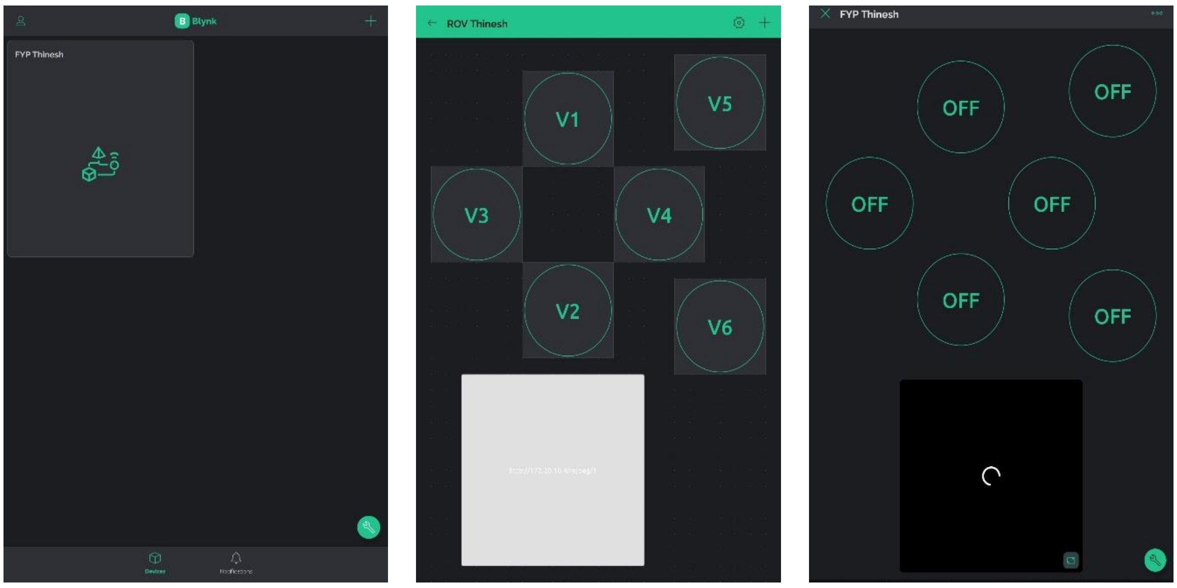
Fig. 7 *Testing ROV for buoyancy, balance, and waterproof (a) Before; (b) After*

3.3 Functionality Test

The ROV undergoes comprehensive functionality tests, focusing on its user-friendly interface, hardware components, and real-time monitoring. The Blynk app and ThingSpeak enable easy command and data management. The ROV's IMU sensor, depth sensor, and thruster are also tested. The serial monitor provides real-time data on functionality and performance. The ESP32 Cam's navigation viewer is also tested, enhancing the ROV's navigational capabilities.

3.3.1 Blynk Configuration

To control the ROV using the Blynk app, users need to create a Blynk account and obtain an authentication token. A new project and template are created within the app, and widgets are added to the interface. A Wi-Fi connection is established, and the NodeMCU ESP8266 microcontroller is programmed to connect to the Blynk server. The Blynk sends control commands to the ROV, which responds. The ROV's capabilities are enhanced by integrating the ESP32-Cam for navigation. The app's success is illustrated in Fig. 8.

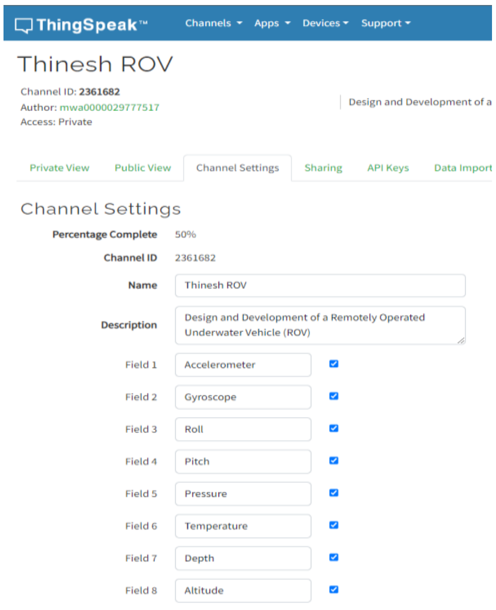


(a) (b) (c)

Fig. 8 *Blynk configuration (a) Device; (b) Widget configuration; (c) Template*

3.3.2 ThingSpeak Configuration

Fig. 9 shows the ThingSpeak configuration for the ROV system includes two channels, one for recording ROV motion and the other for environmental parameters from the IMU-MPU6050 sensor and Bar30 High-Resolution Depth Sensor.



Thinesh ROV

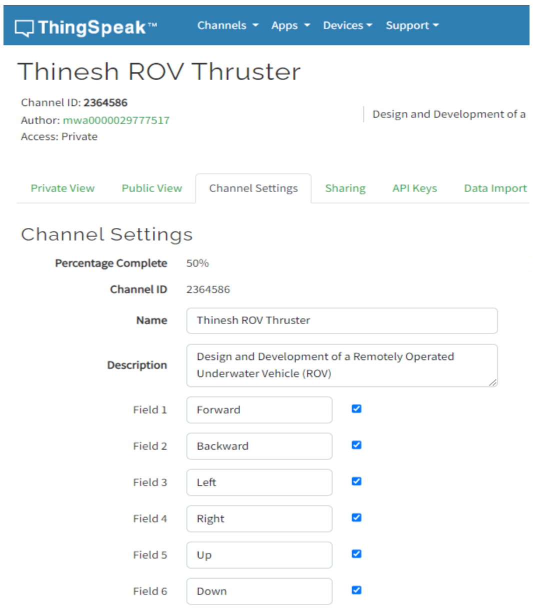
Channel ID: 2361682
Author: mwa0000029777517
Access: Private

Design and Development of a

Channel Settings

Percentage Complete	50%
Channel ID	2361682
Name	Thinesh ROV
Description	Design and Development of a Remotely Operated Underwater Vehicle (ROV)
Field 1	Accelerometer ✓
Field 2	Gyroscope ✓
Field 3	Roll ✓
Field 4	Pitch ✓
Field 5	Pressure ✓
Field 6	Temperature ✓
Field 7	Depth ✓
Field 8	Altitude ✓

(a)



Thinesh ROV Thruster

Channel ID: 2364586
Author: mwa0000029777517
Access: Private

Design and Development of a

Channel Settings

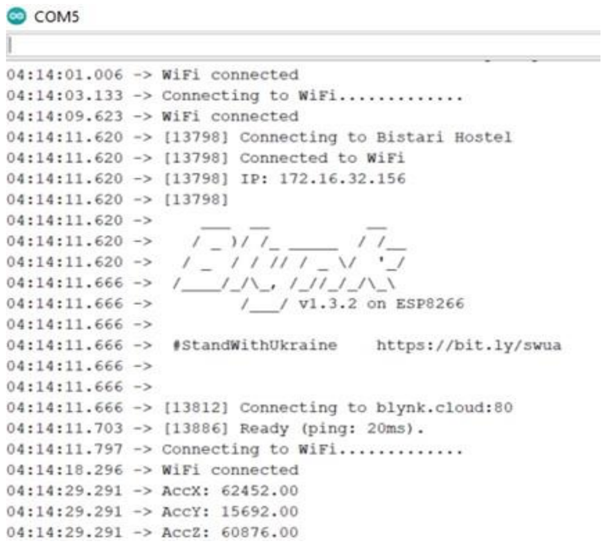
Percentage Complete	50%
Channel ID	2364586
Name	Thinesh ROV Thruster
Description	Design and Development of a Remotely Operated Underwater Vehicle (ROV)
Field 1	Forward ✓
Field 2	Backward ✓
Field 3	Left ✓
Field 4	Right ✓
Field 5	Up ✓
Field 6	Down ✓

(b)

Fig. 9 *ThingSpeak configurations (a) Channel 1; (b) Channel 2*

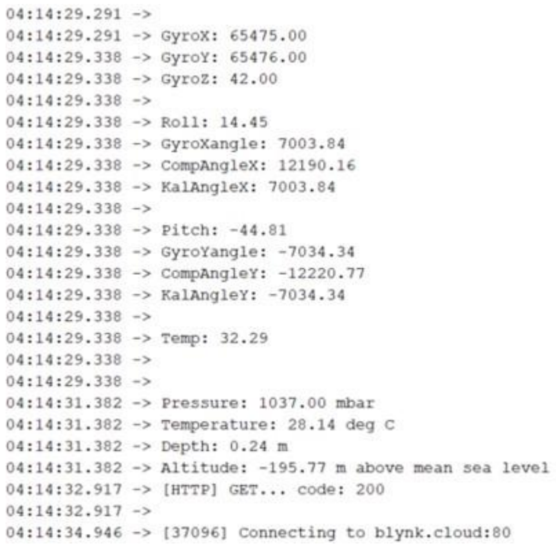
3.3.3 Blynk and ThingSpeak Server Connection with ROV

The testing phase of the system involved verifying the connection between the Blynk app and ThingSpeak server using a ROV. The Arduino IDE serial monitor was used to monitor the ROV's response and data provided to ThingSpeak in real-time. The microcontroller recorded commands sent to the ROV from the Blynk app remote control interfaces, and ThingSpeak cloud system . An example of a successful connection is shown in Fig. 10.



```
04:14:01.006 -> WiFi connected
04:14:03.133 -> Connecting to WiFi.............
04:14:09.623 -> WiFi connected
04:14:11.620 -> [13798] Connecting to Bistari Hostel
04:14:11.620 -> [13798] Connected to WiFi
04:14:11.620 -> [13798] IP: 172.16.32.156
04:14:11.620 -> [13798]
04:14:11.620 ->     ___  __          __
04:14:11.620 ->    / _ )/ /_ _____  / /__
04:14:11.620 ->   / _  / / // / _ \/  '_/
04:14:11.666 ->  /____/_/\_, /_//_/_/\_\
04:14:11.666 ->         /___/ v1.3.2 on ESP8266
04:14:11.666 ->
04:14:11.666 ->  #StandWithUkraine    https://bit.ly/swua
04:14:11.666 ->
04:14:11.666 ->
04:14:11.666 -> [13812] Connecting to blynk.cloud:80
04:14:11.703 -> [13886] Ready (ping: 20ms).
04:14:11.797 -> Connecting to WiFi.............
04:14:18.296 -> WiFi connected
04:14:29.291 -> AccX: 62452.00
04:14:29.291 -> AccY: 15692.00
04:14:29.291 -> AccZ: 60876.00
```

(a)



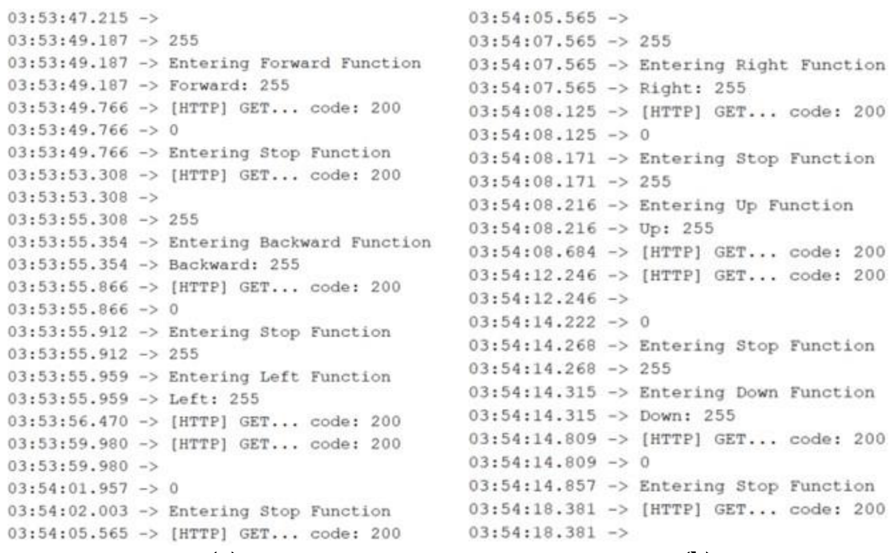
```
04:14:29.291 ->
04:14:29.291 -> GyroX: 65475.00
04:14:29.338 -> GyroY: 65476.00
04:14:29.338 -> GyroZ: 42.00
04:14:29.338 ->
04:14:29.338 -> Roll: 14.45
04:14:29.338 -> GyroXangle: 7003.84
04:14:29.338 -> CompAngleX: 12190.16
04:14:29.338 -> KalAngleX: 7003.84
04:14:29.338 ->
04:14:29.338 -> Pitch: -44.81
04:14:29.338 -> GyroYangle: -7034.34
04:14:29.338 -> CompAngleY: -12220.77
04:14:29.338 -> KalAngleY: -7034.34
04:14:29.338 ->
04:14:29.338 -> Temp: 32.29
04:14:29.338 ->
04:14:29.338 ->
04:14:31.382 -> Pressure: 1037.00 mbar
04:14:31.382 -> Temperature: 28.14 deg C
04:14:31.382 -> Depth: 0.24 m
04:14:31.382 -> Altitude: -195.77 m above mean sea level
04:14:32.917 -> [HTTP] GET... code: 200
04:14:32.917 ->
04:14:34.946 -> [37096] Connecting to blynk.cloud:80
```

(b)

Fig. 10 *Successful connection of ROV with Blynk and ThingSpeak (a) First Data; (b) Second Data*

3.3.4 Thruster Movement

Fig. 11 shows the sent log records commands received by a microcontroller and provides information about the ROV's actions in response to those orders. Command values 255 indicate original operations, while 0 indicates stop commands. This blog is useful for debugging and confirming the success of data transmission to Thing-Speak, as shown by the HTTP GET... code:200.



```
03:53:47.215 ->
03:53:49.187 -> 255
03:53:49.187 -> Entering Forward Function
03:53:49.187 -> Forward: 255
03:53:49.766 -> [HTTP] GET... code: 200
03:53:49.766 -> 0
03:53:49.766 -> Entering Stop Function
03:53:53.308 -> [HTTP] GET... code: 200
03:53:53.308 ->
03:53:55.308 -> 255
03:53:55.354 -> Entering Backward Function
03:53:55.354 -> Backward: 255
03:53:55.866 -> [HTTP] GET... code: 200
03:53:55.866 -> 0
03:53:55.912 -> Entering Stop Function
03:53:55.912 -> 255
03:53:55.959 -> Entering Left Function
03:53:55.959 -> Left: 255
03:53:56.470 -> [HTTP] GET... code: 200
03:53:59.980 -> [HTTP] GET... code: 200
03:53:59.980 ->
03:54:01.957 -> 0
03:54:02.003 -> Entering Stop Function
03:54:05.565 -> [HTTP] GET... code: 200
```

(a)

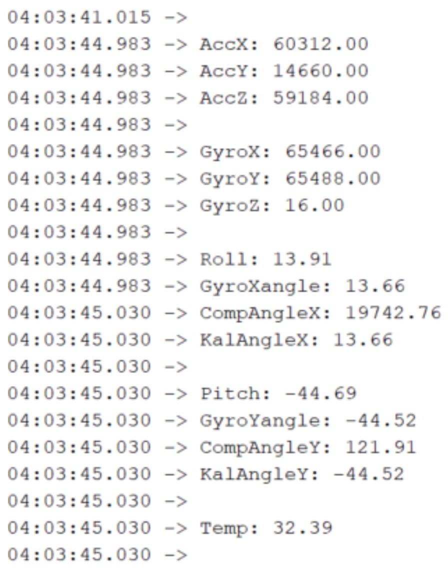
```
03:54:05.565 ->
03:54:07.565 -> 255
03:54:07.565 -> Entering Right Function
03:54:07.565 -> Right: 255
03:54:08.125 -> [HTTP] GET... code: 200
03:54:08.125 -> 0
03:54:08.171 -> Entering Stop Function
03:54:08.171 -> 255
03:54:08.216 -> Entering Up Function
03:54:08.216 -> Up: 255
03:54:08.684 -> [HTTP] GET... code: 200
03:54:12.246 -> [HTTP] GET... code: 200
03:54:12.246 ->
03:54:14.222 -> 0
03:54:14.268 -> Entering Stop Function
03:54:14.268 -> 255
03:54:14.315 -> Entering Down Function
03:54:14.315 -> Down: 255
03:54:14.809 -> [HTTP] GET... code: 200
03:54:14.809 -> 0
03:54:14.857 -> Entering Stop Function
03:54:18.381 -> [HTTP] GET... code: 200
03:54:18.381 ->
```

(b)

Fig. 11 *Thruster movement and data transmission (a) First Data; (b) Second Data*

3.3.5 MPU6050 – IMU Sensor

The IMU Sensor - MPU6050 data, as shown in Fig. 12, has been successfully integrated into the ROV system, providing orientation and motion-related information crucial for underwater navigation and control, demonstrating successful sensor-assisted data-collecting.

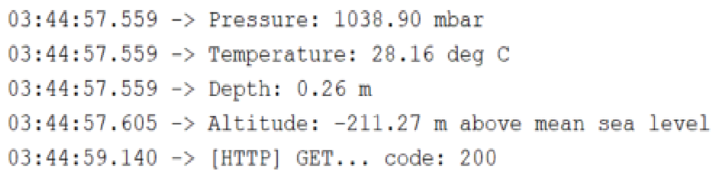


```
04:03:41.015 ->
04:03:44.983 -> AccX: 60312.00
04:03:44.983 -> AccY: 14660.00
04:03:44.983 -> AccZ: 59184.00
04:03:44.983 ->
04:03:44.983 -> GyroX: 65466.00
04:03:44.983 -> GyroY: 65488.00
04:03:44.983 -> GyroZ: 16.00
04:03:44.983 ->
04:03:44.983 -> Roll: 13.91
04:03:44.983 -> GyroXangle: 13.66
04:03:45.030 -> CompAngleX: 19742.76
04:03:45.030 -> KalAngleX: 13.66
04:03:45.030 ->
04:03:45.030 -> Pitch: -44.69
04:03:45.030 -> GyroYangle: -44.52
04:03:45.030 -> CompAngleY: 121.91
04:03:45.030 -> KalAngleY: -44.52
04:03:45.030 ->
04:03:45.030 -> Temp: 32.39
04:03:45.030 ->
```

Fig. 12 *MPU6050 - IMU sensor data*

3.3.6 Bar30 High-Resolution – Depth Sensor

The Depth Sensor - Bar30 High-Resolution examination recorded data on essential underwater parameters like pressure, temperature, depth, and altitude shows in Fig. 13. The data was successfully transmitted to ThingSpeak, confirming the stable connectivity between the platform and the Depth Sensor for examination and monitoring.

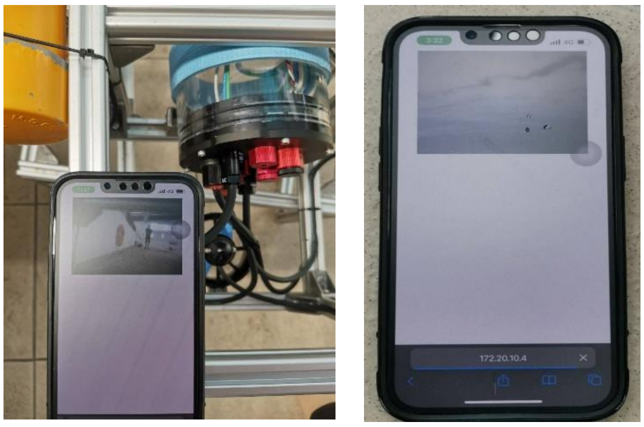


```
03:44:57.559 -> Pressure: 1038.90 mbar
03:44:57.559 -> Temperature: 28.16 deg C
03:44:57.559 -> Depth: 0.26 m
03:44:57.605 -> Altitude: -211.27 m above mean sea level
03:44:59.140 -> [HTTP] GET... code: 200
```

Fig. 13 *Bar30 High-Resolution - Depth sensor data*

3.3.7 ESP32-Cam

The ESP32-Cam navigation camera, as demonstrated in Fig. 14, is reliable and easy to use. Its unique URL link allows direct access to the camera stream, improving user experience. The Blynk interface simplifies management and monitoring, making it an essential component for underwater exploration.



(a) **(b)**

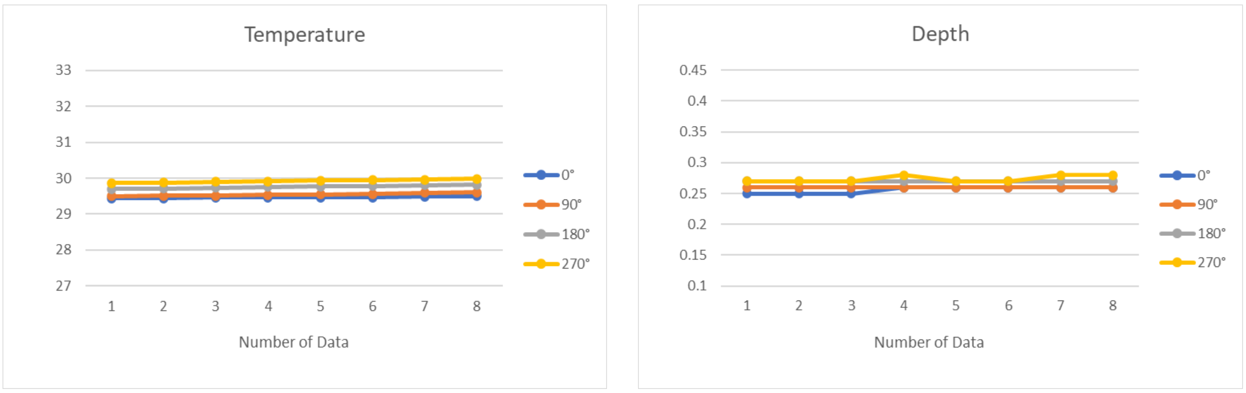
Fig. 14 *Real time monitoring via ESP32-Cam (a) First Data; (b) Second Data*

3.4 Reliability testing

The reliability test evaluates essential sensors accuracy and reliability in various operating conditions and orientations. It includes data orientation analysis, stability analysis 15 minutes before submerging the ROV, and movementes analysis. The MPU6050 and Bar30 High-Resolution depth sensors provide essential data to complete the testing. This comprehensive test ensures the ROV's reliable performance in real-world underwater exploration scenarios.

3.4.1 Data Orientation Analysis

The research project evaluated the performance of a ROV at four different orientations using data from various sensors, aiming to assess its data consistency and responsiveness, providing insights into its stability and dependability in various operating conditions shows in Fig. 15.

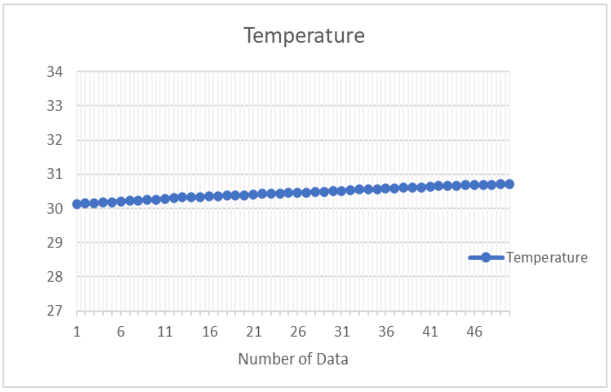


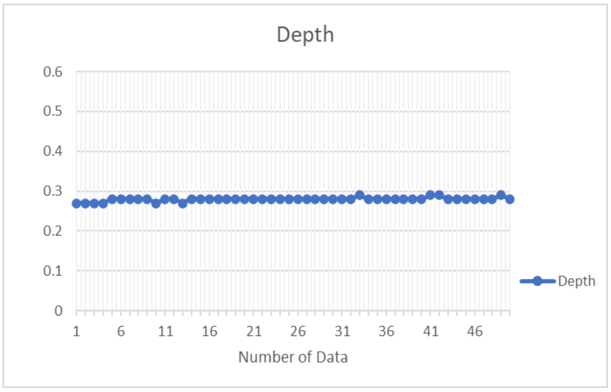
(a) **(b)**

Fig. 15 *Data orientation analysis (a) Temperature; (b) Depth*

3.4.2 Data Stability Analysis

The study evaluates the reliability and continuity of essential metrics recorded by the ROV's sensors in stable settings, using 50 sample data points shows in Fig. 16. It identifies deviations and oscillations, providing information on sensor accuracy.



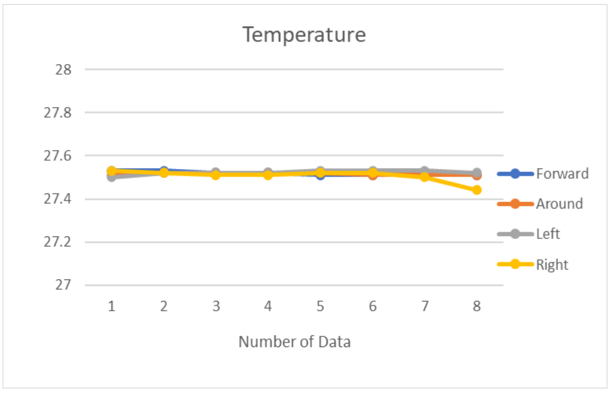


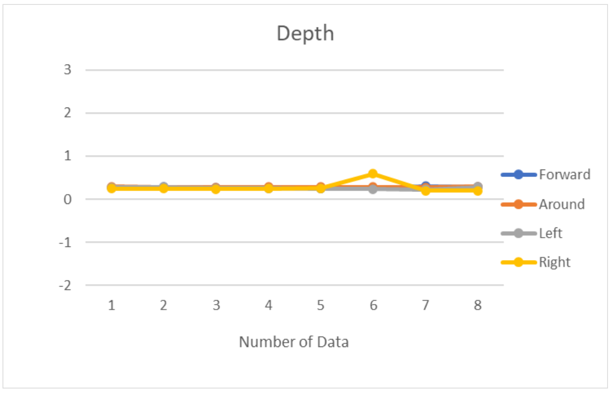
(a) **(b)**

Fig. 16 *Data stability analysis (a) Temperature; (b) Depth*

3.4.3 Data Movements Analysis

The ROV's movements were analyzed using various motions, including Forward, Turn Around, Left, and Right, to evaluate its performance and stability under different navigation conditions, revealing how internal sensors respond to rapid movements shows in Fig. 17.





(a) **(b)**

Fig. 17 *Data movements analysis (a) Temperature; (b) Depth*

4. Conclusion

This study successfully developed a Remotely Operated Underwater Vehicle (ROV), demonstrating its ability to perform underwater tasks. The ROV's performance is enhanced by its user-friendly interface, which can be customized via the Blynk app and ThingSpeak server connection. The ROV's suitability for underwater conditions is confirmed through buoyancy, balancing, and waterproofness tests. Reliability testing provides insights into the ROV's performance under various conditions. The ROV's functionality in submerged environments is improved by incorporating sensors like the ESP32-Cam for navigation, the depth sensor (Bar30 High-Resolution), and the Inertial Measurement Unit (IMU). The ROV's current Wi-Fi connection limitations hinder real-time control and continuous connection. Future research should explore dependable communication alternatives like acoustic modems or RF communication. Improving data transmission protocols and implementing a hybrid communication strategy could enhance operational efficiency and task performance.

Acknowledgement

Communication of this research is made possible through monetary assistance by Universiti Tun Hussein Onn Malaysia and the UTHM Publisher's Office via Publication Fund E15216.

Conflict of Interest

Authors declare that there is no conflict of interests regarding the publication of the paper.

Author Contribution

The author attests to having sole responsibility for the following: planning and designing the study, data collection, analysis and interpretation of the outcomes, and paper writing.

References

[1] Joochim, C., Phadungthin, R., & Srikitsuwan, S. (2016). Design and development of a Remotely Operated Underwater Vehicle. 16th International Conference on Re-search and Education in Mechatronics, REM 2015 – Proceedings, 148–153. https://doi.org/10.1109/REM.2015.7380385

[2] He, Y., Wang, D. B., & Ali, Z. A. (2020). A review of different designs and control models of remotely operated underwater vehicle. Measurement and Control (United Kingdom), 53(9–10), 1561–1570. https://doi.org/10.1177/0020294020952483

[3] Aras, M. S. M., Azis, F. A., Othman, M. N., & Abdullah, S. S. (2012). A Low Cost 4 DOF Remotely Operated Underwater Vehicle Integrated with IMU and Pressure Sensor Keywords: 2012(December), 978–983. https://www.researchgate.net/publication/267752365

[4] Ray, S., Bhowal, R., Patel, P., & Annapurani Panaiyappan, K. (2021). An Overview of the Design and Development of a 6 DOF Remotely Operated Vehicle for Underwater Structural Inspection. ICCISc 2021 - 2021 International Conference on Communication, Control and Information Sciences, Proceedings, 1, 1–6. https://doi.org/10.1109/ICCISc52257.2021.9484879

[5] Azis, F. A., Aras, M. S. M., Rashid, M. Z. A., Othman, M. N., & Abdullah, S. S. (2012). Problem identification for Underwater Remotely Operated Vehicle (ROV): A case study. Procedia Engineering, 41(Iris), 554–560. https://doi.org/10.1016/j.proeng.2012.07.211

[6] Valavanis, K. P., Gracanin, D., Matijasevic, M., Kolluru, R., & Demetriou, G. A. (1997). Control Architectures for Autonomous Underwater Vehicles. IEEE Control Systems, 48–64. https://doi.org/10.1109/37.642974