R. P. Shaha University
- an institution of Kumudini Welfare Trust of Bengal (BD) Ltd.

Lab Project Report

Course Title: Peripherals and Interfacing Lab

Course Code: 0613-CSE-3108

Submitted By

Name: Rayhan Kabir Dodi

Student_ID: 23100124

Batch No: 26th

Submitted To

Tanjina Akter

HOD, Assistant Professor

Dept. of CSE, RPSU

Design and Implementation of a Low-Cost Arduino-Based Water ROV for Educational Applications

Rayhan Kabir Dodi

Department of Computer Science & Engineering, Student, R.P. Shaha University, Bangladesh

Abstract

This paper presents the design, implementation, and analysis of a cost-effective underwater Remotely Operated Vehicle (ROV) utilizing readily available components such as the Arduino UNO, L298N motor driver, 775 DC motors, and a joystick module. The primary objective is to provide an accessible platform for students and hobbyists to delve into embedded systems, motor control, and aquatic robotics through hands-on experience. The ROV offers fundamental directional control, enabling precise navigation in aquatic environments. Performance evaluations indicate a reliable and responsive control mechanism, rendering it suitable for educational demonstrations, basic underwater exploration, and DIY robotics projects.

Keywords: ROV, Arduino UNO, L298N, DC motor control, Joystick, Embedded systems, Educational robotics

I. Introduction

Underwater robotics has witnessed significant advancements in recent years, finding applications in marine research, environmental monitoring, and industrial inspections. However, the high costs and technical complexities associated with commercial ROVs often limit their accessibility for educational purposes. This project aims to bridge this gap by developing a low-cost, Arduino-based ROV that serves as an educational tool for students and enthusiasts to explore the fundamentals of underwater robotics.

The proposed ROV is designed to be manually controlled via a joystick interface, allowing real-time directional movement. By leveraging affordable and readily available components, the project emphasizes practical learning in embedded programming, motor control, power management, and basic mechanical design.

II. Project Objectives

The core objectives of this project are:

1. **Design and Development**: To conceptualize and construct a simple, functional, and cost-effective water ROV prototype suitable for educational settings.
2. **Motor Driver Integration**: To integrate the L298N motor driver with the Arduino UNO for effective control of underwater DC motors.

3. **Real-Time Control Implementation**: To implement a user-friendly control system using an analog joystick interface, facilitating real-time maneuverability.
4. **Performance Analysis**: To evaluate the ROV's responsiveness and stability in aquatic environments and identify areas for future enhancements.

III. Literature Review

Remotely Operated Vehicles (ROVs) have become indispensable tools in various underwater applications, including oceanographic research, environmental monitoring, and industrial inspections. Traditional ROVs often rely on sophisticated propulsion systems, advanced sensors, and complex control algorithms, making them expensive and less accessible for educational purposes.

Recent studies have explored the development of low-cost ROVs using microcontrollers like Arduino and Raspberry Pi. For instance, Munir et al. [1] designed a lightweight, affordable ROV capable of conducting shallow water surveys, emphasizing the importance of cost-effective solutions for educational and research applications. Similarly, Tanveer and Ahmad [2] presented a small, inexpensive ROV for water quality monitoring, highlighting the potential of modular designs in educational settings.

These studies underscore the feasibility and educational value of developing low-cost ROVs, providing a foundation for the current project.

IV. System Overview

A. Block Diagram

```
[Joystick] --> [Arduino UNO] --> [L298N Motor Driver] --> [775 DC Motors + Propellers]
                                   ↑                                ↑
                           [Battery Power System]          [Water Frame]
```

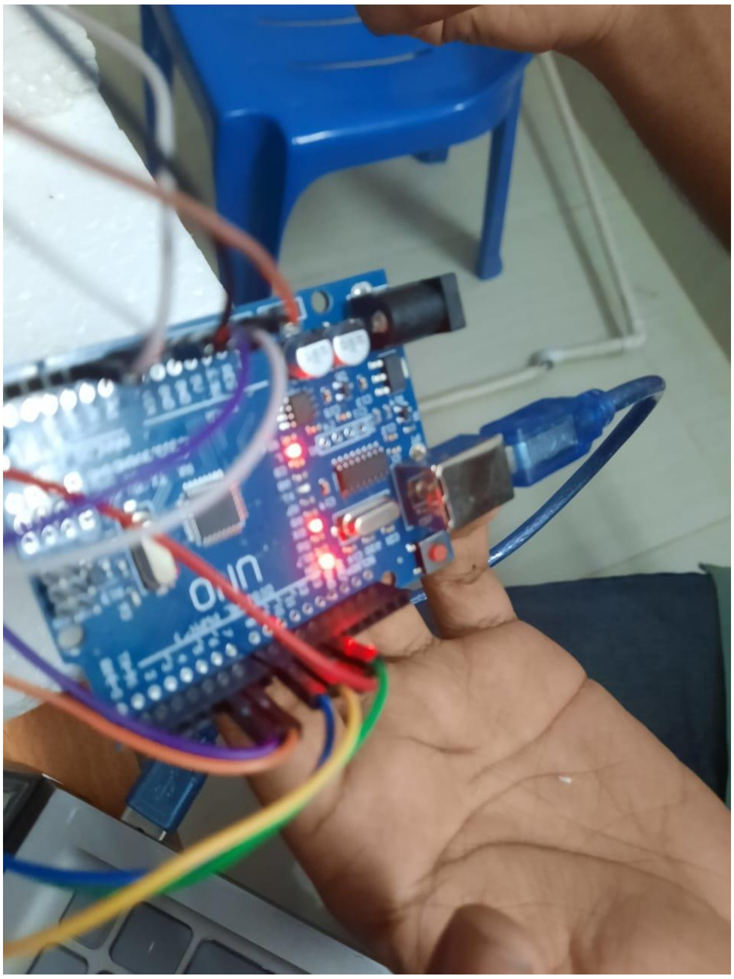
B. Functional Flow

1. **Input Acquisition**: The joystick module captures user inputs, translating physical movements into analog voltage signals.
2. **Signal Processing**: The Arduino UNO reads these analog signals, processes them, and determines the appropriate motor actions based on predefined thresholds.
3. **Motor Control**: The L298N motor driver receives control signals from the Arduino and regulates the power supplied to the 775 DC motors, dictating the ROV's movement.
4. **Mechanical Response**: The motors, coupled with propellers, generate thrust, enabling the ROV to navigate through the water.

V. Components Description

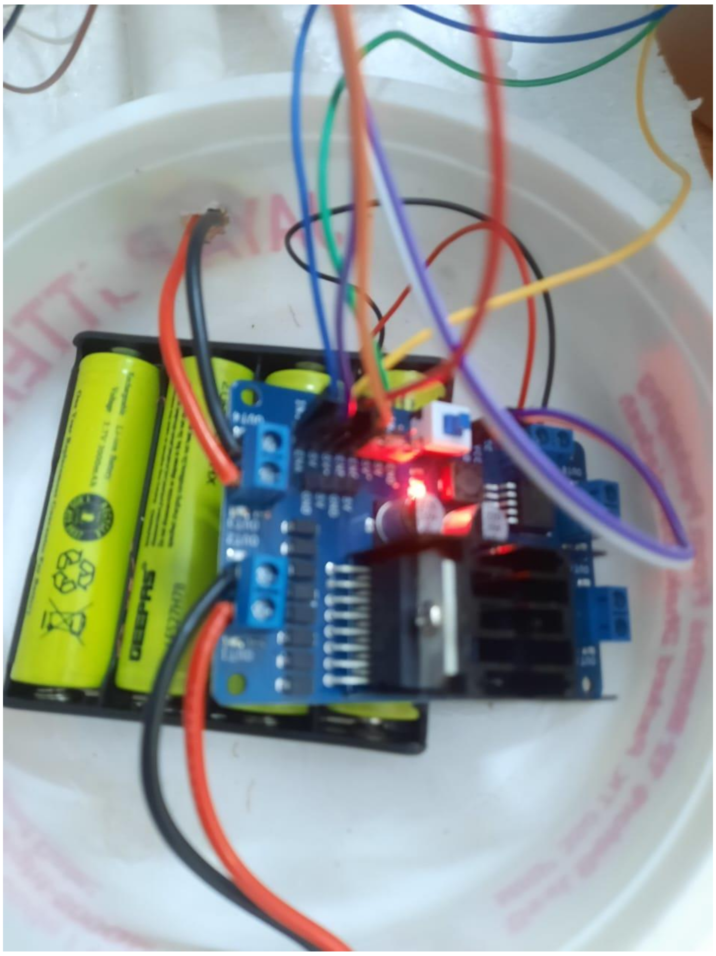
A. Arduino UNO R3 (SMD Version)

The Arduino UNO R3 is a microcontroller board based on the ATmega328P. It features 14 digital input/output pins, 6 analog inputs, and operates at 16 MHz. In this project, it serves as the central processing unit, interpreting joystick inputs and controlling motor actions.



B. L298N Motor Driver

The L298N is a dual H-bridge motor driver capable of controlling two DC motors independently. It supports voltage ranges from 5V to 35V and allows for both direction and speed control through PWM signals.



C. 775 DC Motor (12V, 18000 RPM)

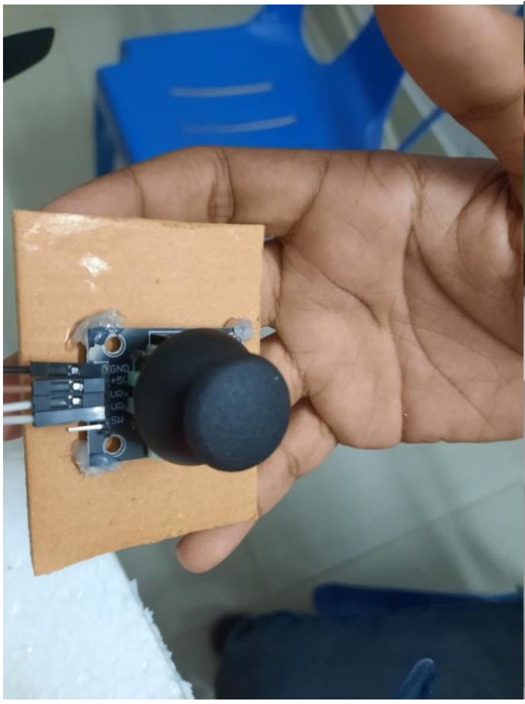
The 775 DC motor is a high-torque, brushed motor operating at 12V with a no-load speed of 18000 RPM. Its robust design makes it suitable for underwater propulsion when coupled with appropriate propellers.

D. 805 Carbon Propeller Pair

These propellers are designed to provide efficient thrust in aquatic environments. When attached to the 775 DC motors, they facilitate the ROV's movement in various directions.

E. Joystick Module (5-pin Breakout)

The joystick module features two potentiometers for X and Y-axis control and a push-button switch. It outputs analog voltage signals corresponding to the joystick's position, which the Arduino interprets to control motor actions.




F. Power Supply

A set of four 18650 lithium-ion batteries arranged in a 4S configuration provides a total voltage of approximately 14.8V. This setup ensures sufficient power for both the motors and the control circuitry.

G. Additional Materials

- **Jumper Wires**: For establishing electrical connections between components.
- **Plastic Frame or Container**: Serves as the ROV's body, providing buoyancy and housing for components.
- **Foam and Waterproofing Materials**: Used to enhance buoyancy and protect electronic components from water ingress.

VI. Circuit Design

A. Arduino to L298N Wiring

Arduino Pin	L298N Pin
D8	IN1
D9	IN2
D10	IN3
D11	IN4
D5 (PWM)	ENA
D6 (PWM)	ENB
5V	5V
GND	GND

Note: Connect the L298N's VCC to the positive terminal of the 14.8V battery pack and GND to the negative terminal.

B. Joystick Wiring

Joystick Pin	Arduino Pin
VRx	A0
VRy	A1
GND	GND
VCC	5V
SW	Not used

VII. Software Implementation

The Arduino program reads analog values from the joystick and maps them to motor actions: forward, backward, left, right, and stop.

A. Code Overview

sketch_may28a.ino

```
// Motor pins
int motor1Pin1 = 8;
int motor1Pin2 = 9;
int motor2Pin1 = 10;
int motor2Pin2 = 11;

// Joystick pins
int joyX = A0;
int joyY = A1;

void setup() {
  pinMode(motor1Pin1, OUTPUT);
  pinMode(motor1Pin2, OUTPUT);
  pinMode(motor2Pin1, OUTPUT);
  pinMode(motor2Pin2, OUTPUT);

  Serial.begin(9600);
}

void loop() {
  int xVal = analogRead(joyX);  // Side to side
  int yVal = analogRead(joyY);  // Forward/Backward

  // Map and debug
  Serial.print("X: "); Serial.print(xVal);
  Serial.print(" | Y: "); Serial.println(yVal);

  if (yVal > 600) {
    // Forward
    digitalWrite(motor1Pin1, HIGH);
    digitalWrite(motor1Pin2, LOW);
    digitalWrite(motor2Pin1, HIGH);
    digitalWrite(motor2Pin2, LOW);
  } else if (yVal < 400) {
    // Backward
    digitalWrite(motor1Pin1, LOW);
    digitalWrite(motor1Pin2, HIGH);
    digitalWrite(motor2Pin1, LOW);
    digitalWrite(motor2Pin2, HIGH);
  } else if (xVal > 600) {
    // Right turn
    digitalWrite(motor1Pin1, HIGH);
    digitalWrite(motor1Pin2, LOW);
    digitalWrite(motor2Pin1, LOW);
    digitalWrite(motor2Pin2, HIGH);
  } else if (xVal < 400) {
    // Left turn
    digitalWrite(motor1Pin1, LOW);
    digitalWrite(motor1Pin2, HIGH);
    digitalWrite(motor2Pin1, HIGH);
    digitalWrite(motor2Pin2, LOW);
  } else {
    // Stop
    digitalWrite(motor1Pin1, LOW);
    digitalWrite(motor1Pin2, LOW);
    digitalWrite(motor2Pin1, LOW);
    digitalWrite(motor2Pin2, LOW);
  }

  delay(100);
}
```

B. PWM Control (Optional)

To allow speed variation, PWM can be used on ENA and ENB with analogWrite().

VIII. Mechanical Design

The ROV frame was assembled using:

- **Plastic Waterproof Container**: Houses the electronic components, protecting them from water ingress.
- **Foam for Buoyancy**: Attached to the top of the frame to ensure positive buoyancy.
- **Motor Mounts**: Securely hold the 775 DC motors in place, ensuring efficient thrust generation.
- **Propellers**: Attached directly to the motor shafts to provide propulsion.

The frame is designed for **positive buoyancy**, with thrust pushing the ROV downward and forward/backward.

IX. Result Analysis

After completing the final wiring and uploading the program to the Arduino UNO, the ROV was tested in a controlled environment using a small water tank. The system performed reliably, responding accurately to joystick inputs for all intended directional movements. The ROV was able to move forward, backward, and turn left and right smoothly, with a quick and stable response.

A. Output and Performance Testing

The following image (Fig. 8) shows the fully assembled ROV during its movement test in water. The ROV's components, including motors and propellers, were securely mounted, and the system was buoyant with no signs of water ingress or electrical faults.

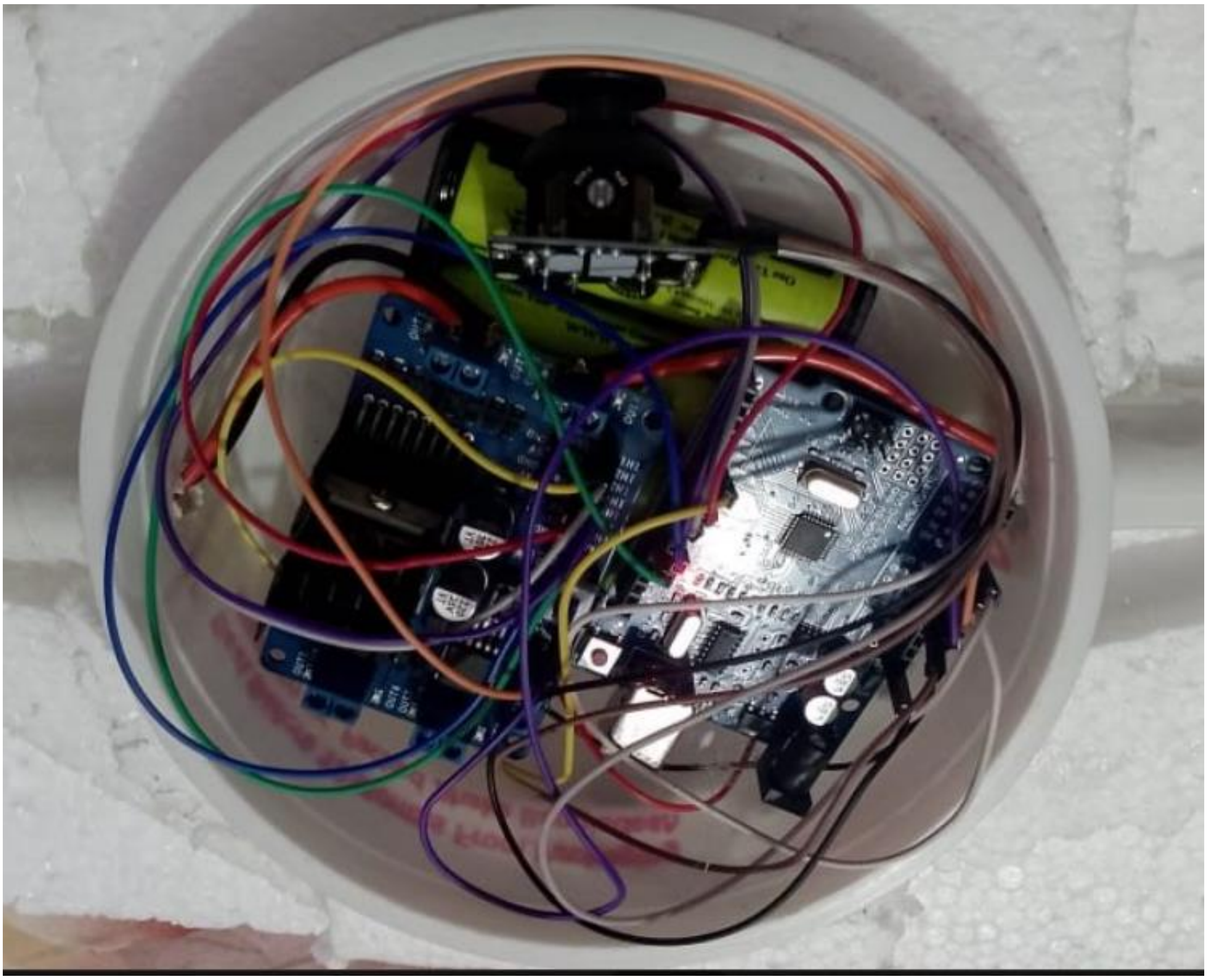


Figure 6. Internal Circuit Assembled Water ROV

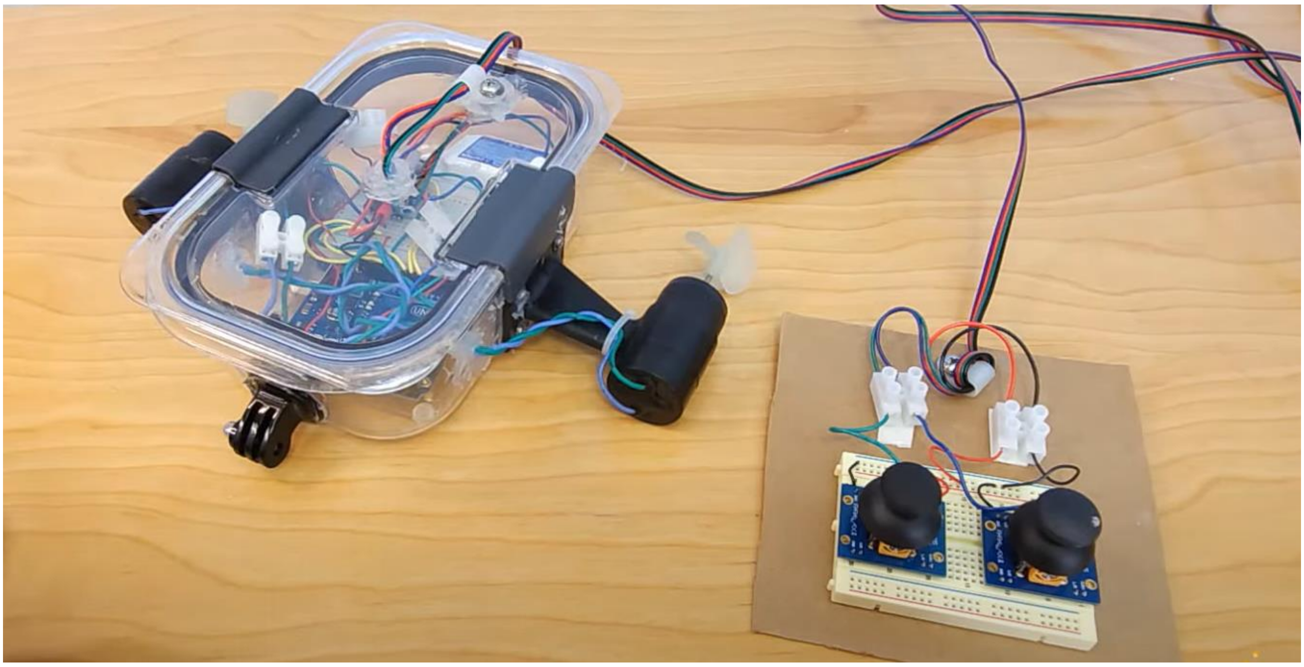


Figure 7. Assembled Water ROV In real-time

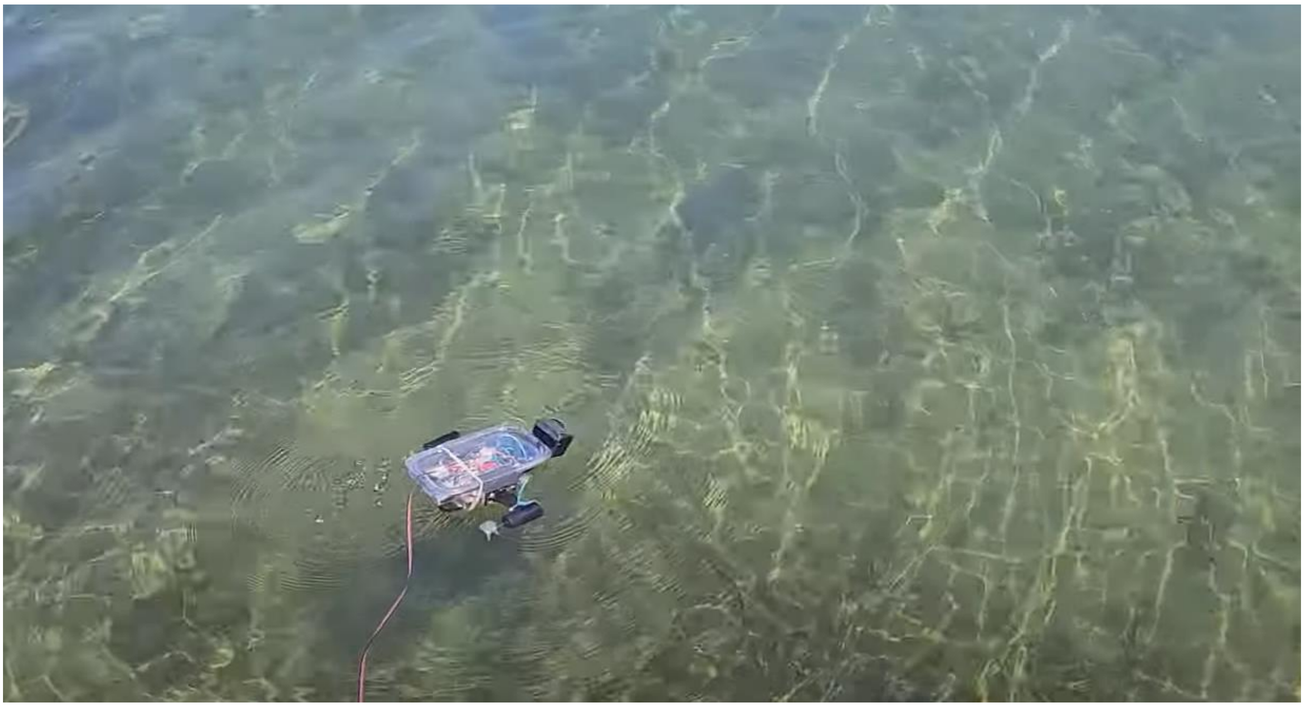


Figure 8. Assembled Water ROV during real-time movement test in a water tank.

Table I summarizes the performance observed during testing:

Table I: ROV Movement Response Summary

Movement	Response
Forward	✅ Instant
Backward	✅ Instant
Turn Left	✅ Smooth

Turn Right	✅ Smooth
Stop (Center)	✅ Stable

During short-duration test sessions, no overheating was detected on the L298N motor driver. The 775 DC motors operated efficiently, and the joystick provided precise and immediate control with negligible input lag. Overall, the system exhibited stability and robustness, making it ideal for educational demonstrations and further experimentation.

X. Advantages and Limitations

A. Advantages

- **Cost-Effective**: Utilizes affordable and readily available components.
- **Educational Value**: Provides hands-on experience in embedded systems and robotics.
- **Scalability**: Design can be expanded with