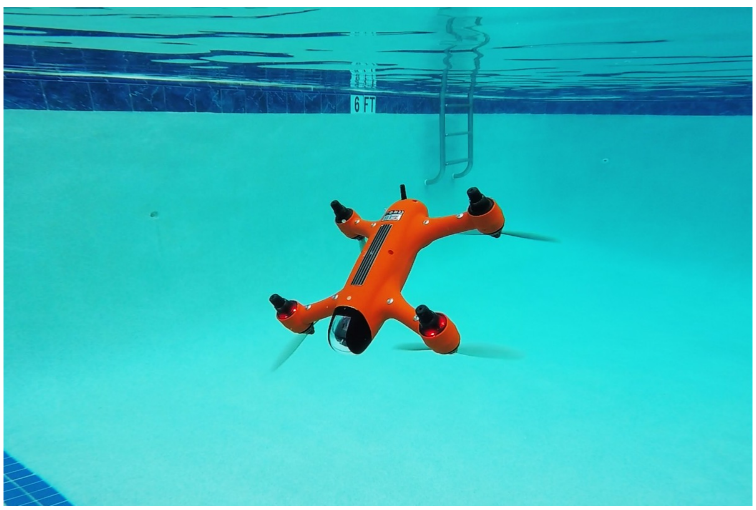
Department of Computer Science & Engineering
The University of Texas at Arlington

Detailed Design Specification
CSE 4317: Senior Design II
Spring 2022



The Drowning Robots
Ocean Debris Cleanup Bot

Hunter Redhead
Apar Pokhrel
Joanne Mathew
Sean Walter

Revision History

Revision	Date	Author(s)	Description
1.0	3.13.2022	HR, AP, JM, SW	first draft

Contents

List of Figures

List of Tables

1 Introduction

The final outcome of the project is an underwater robot capable of performing a set of required tasks designated in the 2022 IEEE Robotics Competition. This section provides a high-level overview of the desired functionalities, the methods of use, and the targeted audience.

The device shall perform a series of tasks which includes the following. First, the robot shall maneuver through the underwater rings to pick up a block, transport it back through the same rings, and deposit on an underwater shelf. Second, the robot shall push a button to release tennis balls from a box, and then scoop as many tennis balls as it can from the water's surface and drop it in a container. The device will complete these tasks by employing the use of a multi-functional component handler in conjunction with a user-controlled fine-movement system which combines the use of thrusts and ballast tanks.

The design structure involves four main layers: External Component System, HMI/Controller system, Sensors system, and Movement System, each of which are tasked with different functionalities and equipped with a varied level of capabilities.

2 System Overview

The Human Interface/Controller layer allows the user to interact with the mechanical design and the functionalities of the underwater ROV. Similarly, the Sensor System is responsible for collecting data from the environment around the robot. It will stream data out for the pilot or control unit to use in the movement of the robot. The Movement System consists of the parts involved for movement. For us, that will be the thruster and ballast tank subsystems. These subsystems will receive input from the Human Machine Interface/Controller which will control the power levels on the thrusters and pumps regulating the movement of the robot. The External Component System will be used to perform required tasks for the ROV, such as picking up and transporting the block and retrieving the tennis balls from the water surface.

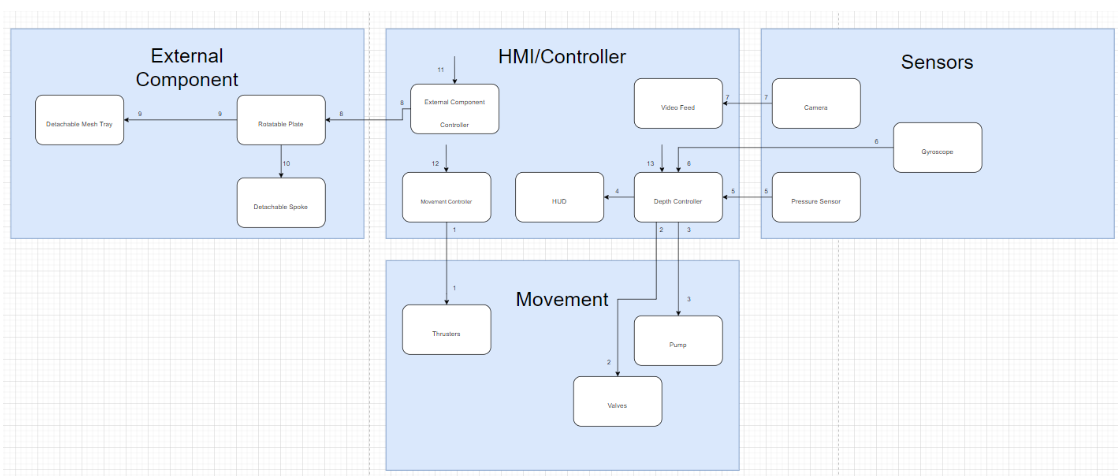


Figure 1: System Architecture

3 Human Interface / Controller

The Human Interface / Controller layer is the layer responsible for taking user input, delivering it to the robot and using that data to interact with the hardware. It is also responsible for feeding data such as video data from the robot to the user. It uses a React JS webpage and a Node JS backend.

3.1 Layer Hardware

To host the webpage used for the controller, any computer that can run a webpage is viable. As for the backend, we are using a Raspberry Pi (X) Model B to host the backend on the robot so that it can interact with the hardware systems of the robot.

3.2 Layer Operating System

The Raspberry Pi (X) Model B is going to use the Raspbian Linux distribution of Linux. For hosting the webapp controller we will be using a laptop with Windows 10 or MacOS.

3.3 Layer Software Dependencies

- Node JS (v16.13.2): Framework for Backend
- React JS (v17.0.2): Framework for Controller
- Axios (v0.25.0): HTTP API for the Controller to send HTTP messages
- Express (v4.17.2): HTTP API for the Backend to receive HTTP messages
- Cors (v2.8.5): For allowing CORS HTTP messages to reach the Backend
- Body-parser (v1.19.1): For parsing the HTTP messages into JSON objects

3.4 External Component Controller

The UI controls that are responsible for changing the state of the external component (grabby thingy).

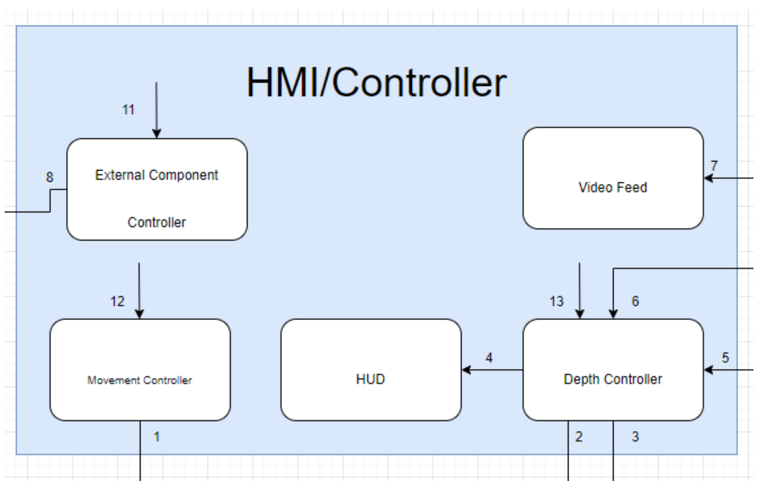


Figure 2: HMI Subsystem Diagram

3.4.1 Subsystem Programming Languages

Javascript, HTML, and CSS for the implementation of the button on the web app.

3.4.2 Subsystem Data Structures

HTTP POST Message packet with boolean state value that represents the state of the external component being passed into the POST body.

3.5 Movement Controller

The UI controls that are responsible for taking the user input for movement commands to the robot.

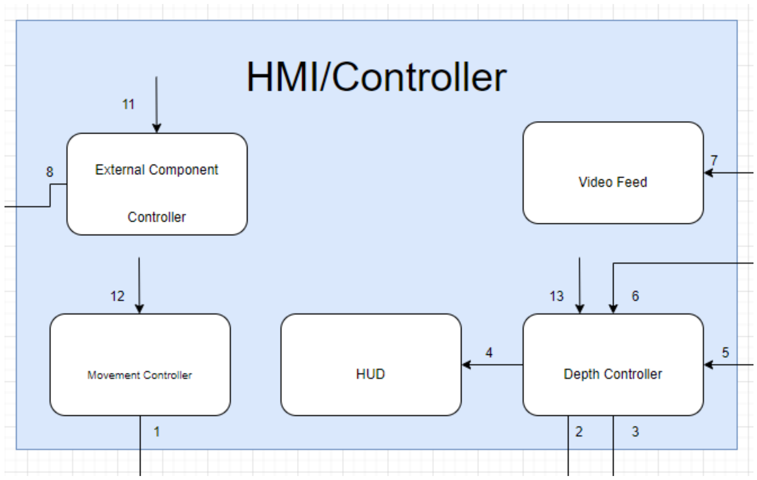


Figure 3: HMI Subsystem Diagram

3.5.1 Subsystem Software Dependencies

Software dependencies for this subsystem have not yet been determined. This section will be updated in the future.

3.5.2 Subsystem Programming Languages

Javascript to capture the events of key presses from the keyboard.

3.5.3 Subsystem Data Structures

HTTP POST Message packet with two values in the body. An action, a string used for informing what kind of action it is, and a direction, a character that represents the direction the robot is to move in.
Movement Actions:

- "start" = Turn on thrusters in x direction
- "stop" = Turn off thrusters in x direction

Movement Directions:

- w = forward
- s = backward
- a = left
- d = right

3.6 Depth Controller

The UI slider the user uses to control the depth of the robot.

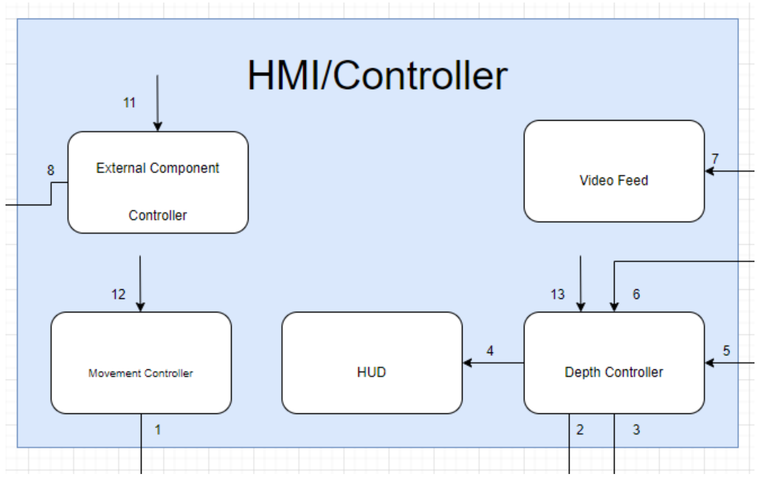


Figure 4: HMI Subsystem Diagram

3.6.1 Subsystem Programming Languages

HTML, CSS, and Javascript to handle the implementation of the slider for the web app.

3.6.2 Subsystem Data Structures

HTTP POST Message packet with two values in the body. An action, a string used for informing what kind of action it is (in this case action = "depth"), and a depth, a floating point number representing the depth the user wants the robot to go to.

3.7 Video Feed

Video being streamed from the camera on the robot to the web app for the user to see what the robot is doing.

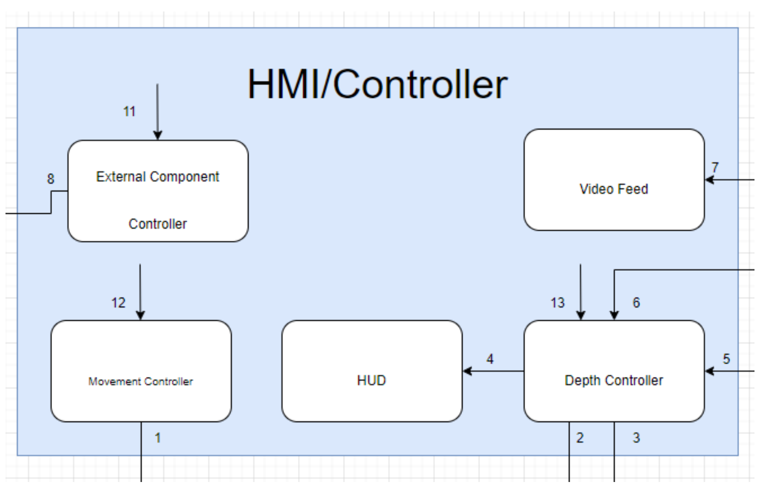


Figure 5: HMI Subsystem Diagram

3.7.1 Subsystem Hardware

Video Camera being used for the robot.

3.7.2 Subsystem Software Dependencies

Currently do not know if any particular software dependencies will be used for displaying the video feed.

3.7.3 Subsystem Programming Languages

HTML and CSS for the displaying the video block on the web app. Javascript used for getting the video data from the backend and streaming it to the webpage.

3.7.4 Subsystem Data Structures

Currently do not know if any particular data structures will be used.

3.7.5 Subsystem Data Processing

Currently do not know how data will be processed.

3.8 HUD

The display that shows vital information for accurately and correctly controlling the robot. In this case, it is primarily the true depth the robot is at.

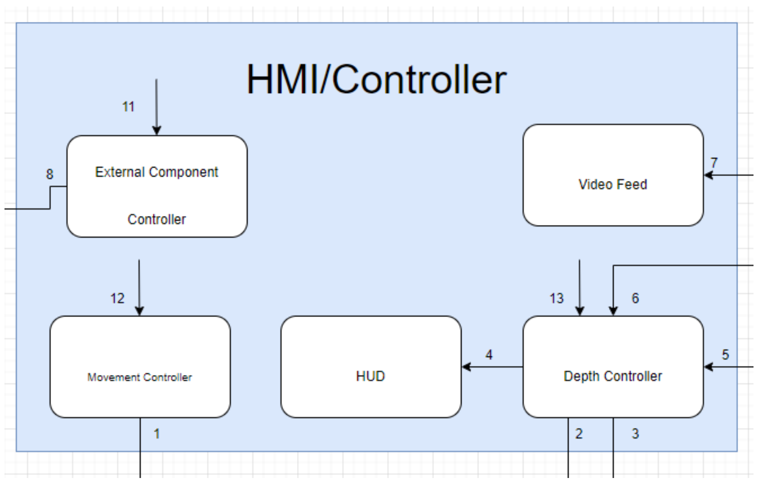


Figure 6: HMI Subsystem Diagram

3.8.1 Subsystem Programming Languages

HTML and CSS for displaying the HUD information in a viable format for the user. Javascript is to be used to interpret the data received from the instrument that measures the depth.

3.8.2 Subsystem Data Structures

Do not currently know if any particular data structures will be necessary.

3.8.3 Subsystem Data Processing

Do not currently know how the data will be processed.

4 Sensor System

The sensor system is composed of three main subsystems: the camera system, the gyroscope system, and the pressure/depth sensor system. The camera system is responsible for providing the user with live video from both the front and back of the robot. The gyroscope system is responsible for letting the user know the orientation of the ROV. The pressure/depth sensor system is responsible for letting the user know how deep the ROV is and works in tandem with the gyroscope system to help orient the ROV.

4.1 Camera System

The camera system is responsible for providing the user with live video feeds from both the front and the back of the ROV.

4.1.1 Subsystem Hardware

The system will consist of a camera chip that outputs raw data which can be processed to provide a live video feed.

4.1.2 Subsystem Software Dependencies

The software dependencies for this subsystem have not yet been determined. This section will be updated in the future.

4.1.3 Subsystem Programming Languages

Code for this subsystem will either be written in C or in Arduino code.

4.1.4 Subsystem Data Structures

Data structures for this subsystem have not yet been developed. This section will be updated in the future.

4.1.5 Subsystem Data Processing

Data processing methods for this subsystem have not yet been developed. This section will be updated in the future.

4.2 Gyroscope System

The gyroscope system is responsible for providing the user with orientation data for the ROV.

4.2.1 Subsystem Hardware

The system will consist of a combined gyroscope and accelerometer chip that outputs raw data which can be processed to provide orientation readings for the ROV to the user.

4.2.2 Subsystem Software Dependencies

The software dependencies for this subsystem have not yet been determined. This section will be updated in the future.

4.2.3 Subsystem Programming Languages

Code for this subsystem will either be written in C or in Arduino code.

4.2.4 Subsystem Data Structures

Data structures for this subsystem have not yet been developed. This section will be updated in the future.

4.2.5 Subsystem Data Processing

Data processing methods for this subsystem have not yet been developed. This section will be updated in the future.

4.3 Pressure/Depth Sensor System

The pressure/depth sensor system is responsible for providing the user with data concerning the depth of the ROV once submerged.

4.3.1 Subsystem Hardware

The system will consist of a pressure/depth sensor chip that outputs raw data which can be processed to provide live pressure/depth readings to the user.

4.3.2 Subsystem Software Dependencies

The software dependencies for this subsystem have not yet been determined. This section will be updated in the future.

4.3.3 Subsystem Programming Languages

Code for this subsystem will either be written in C or in Arduino code.

4.3.4 Subsystem Data Structures

Data structures for this subsystem have not yet been developed. This section will be updated in the future.

4.3.5 Subsystem Data Processing

Data processing methods for this subsystem have not yet been developed. This section will be updated in the future.

5 Movement System

We will be building a ballast tank system with the use of thrusters. The ballast tank system will include four varying valves and a pump with PVC pipes acting as the tanks. For the thrusters we will have one on either side to provide forward movement. We will also be using an Arduino board and the on-board microcontroller.

5.1 Ballast Tank System

The ballast tank system is responsible for submerging, surfacing, and stabilizing the underwater movements of the ROV.

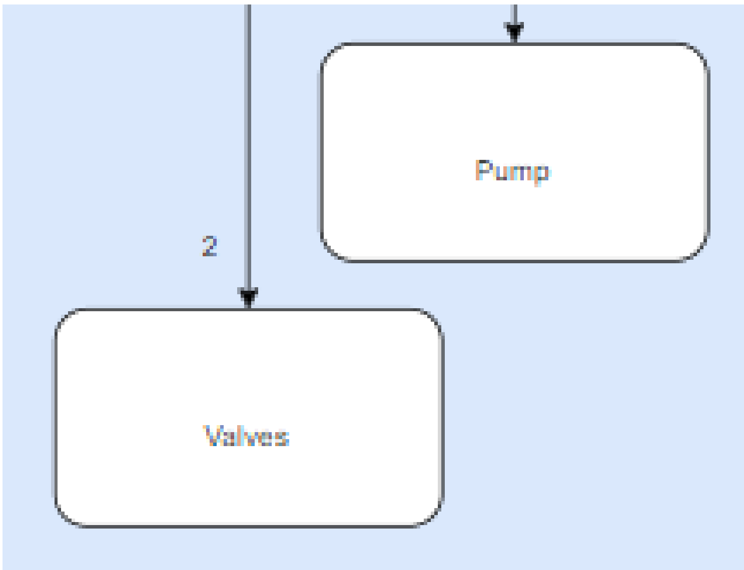


Figure 7: Both pump and valves control the system

5.1.1 Subsystem Hardware

- One pump ()
- Two 1 by 2 valves (uxcell Plastic Water Electric Solenoid Valve)
- Two 1 by 4 valves (uxcell Plastic Water Electric Solenoid Valve)
- Piping ()
- Arduino MEGA 2560

5.1.2 Subsystem Software Dependencies

The software dependencies for this subsystem have not yet been determined. This section will be updated in the future.

5.1.3 Subsystem Programming Languages

Code for this subsystem will either be written in C or in Arduino code.

5.1.4 Subsystem Data Structures

Data structures for this subsystem have not yet been developed. This section will be updated in the future.

5.1.5 Subsystem Data Processing

Data processing methods for this subsystem have not yet been developed. This section will be updated in the future.

5.2 Thrusters System

This system is the four thrusters that control the forwards, backwards and turning movements of our ROV. It is comprised of both clockwise and counter clockwise thrusters.

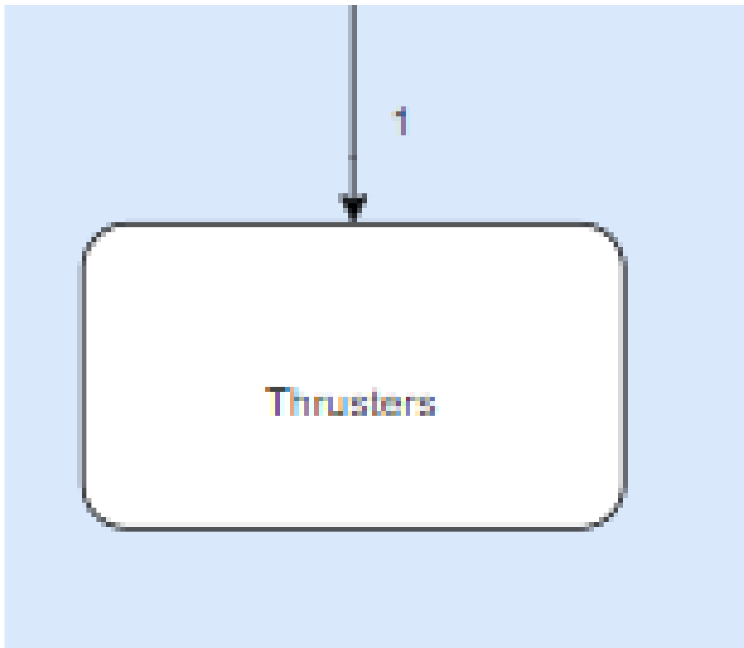


Figure 8: This system mostly just has thrusters

5.2.1 Subsystem Hardware

- Two counter clockwise thrusters ()
- Two clockwise thrusters ()
- Arduino MEGA 2560

5.2.2 Subsystem Software Dependencies

The software dependencies for this subsystem have not yet been determined. This section will be updated in the future.

5.2.3 Subsystem Programming Languages

Code for this subsystem will either be written in C or in Arduino code.

5.2.4 Subsystem Data Structures

Data structures for this subsystem have not yet been developed. This section will be updated in the future.

5.2.5 Subsystem Data Processing

Data processing methods for this subsystem have not yet been developed. This section will be updated in the future.

6 External Component Handler System

The component handler is composed of a spike for picking up the block and a motor controlled net for scooping up the tennis balls.

6.1 Spike

The spike will be composed of a long metal rod.

6.1.1 Subsystem Hardware

The spike will be composed of a long metal rod that can be slipped through the eyehole on top of the block.

6.2 Motor Controlled Net

A motor will be used to turn a net that can be used in combination with the submerging and surfacing capabilities of the ROV to scoop up the tennis balls and dump them into the trash receptacle.

6.2.1 Subsystem Hardware

A U-shaped PVC pipe will be attached to the spoke which will go through the center shaft of the motor. The netting will stretch across PVC piping attached to the main PVC pipe connected to the spike.

6.2.2 Subsystem Software Dependencies

The software dependencies for this subsystem have not yet been determined. This section will be updated in the future.

6.2.3 Subsystem Programming Languages

Code for this subsystem will either be written in C or in Arduino code.

6.2.4 Subsystem Data Structures

Data structures for this subsystem have not yet been developed. This section will be updated in the future.

6.2.5 Subsystem Data Processing

Data processing methods for this subsystem have not yet been developed. This section will be updated in the future.

7 APPENDIX A

This appendix will be updated in the future.

References