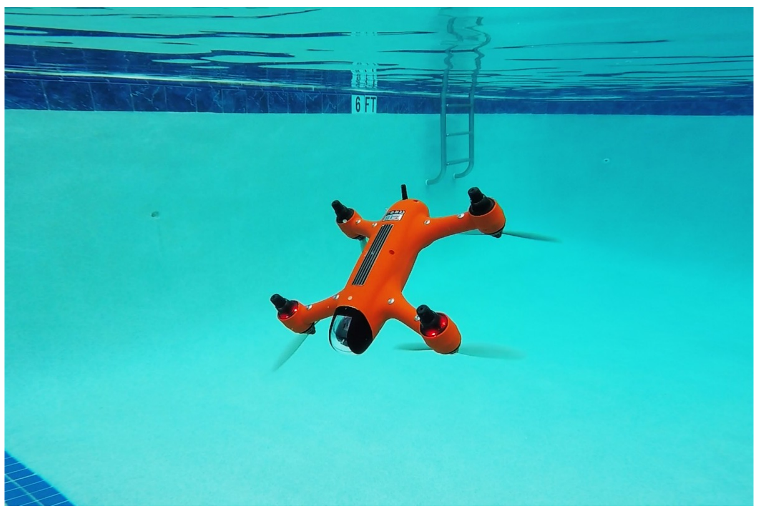
# Department of Computer Science & Engineering
# The University of Texas at Arlington


# Architectural Design Specification
# CSE 4316: Senior Design I
# Fall 2021




# The Drowning Robots
# Ocean Debris Cleanup Bot


**Joanne Mathew**
**Apar Pokhrel**
**Hunter Redhead**
**Sean Walter**

# REVISION HISTORY

| Revision | Date | Author(s) | Description |
|---|---|---|---|
| 0.1 | 12.3.2021 | JM, AP, HR, SW | document creation |
| 1.0 | 12.08.2021 | JM, AP, HR, SW | document completion |
| 1.1 | 5.04.2022 | JM, AP, HR, SW | document update |

# Contents

## List of Figures



## List of Tables

# 1 INTRODUCTION

The final product will consist of a device capable of executing required tasks in the 2022 IEEE Robotics Competition. The purpose of this section is to provide a high-level overview of desired device functionality; the method of use and intended audience for this device will be outlined in the subsections below. The device should be used to perform the two required competition tasks, the first of which involves moving through underwater rings to pick up a block, transport the block back through the rings, and deposit the block on an underwater shelf, and the second of which involves pushing a button to release tennis balls from a box, and then obtaining as many of these tennis balls from the water's surface as possible and dropping them into 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. T

# 2 System Overview

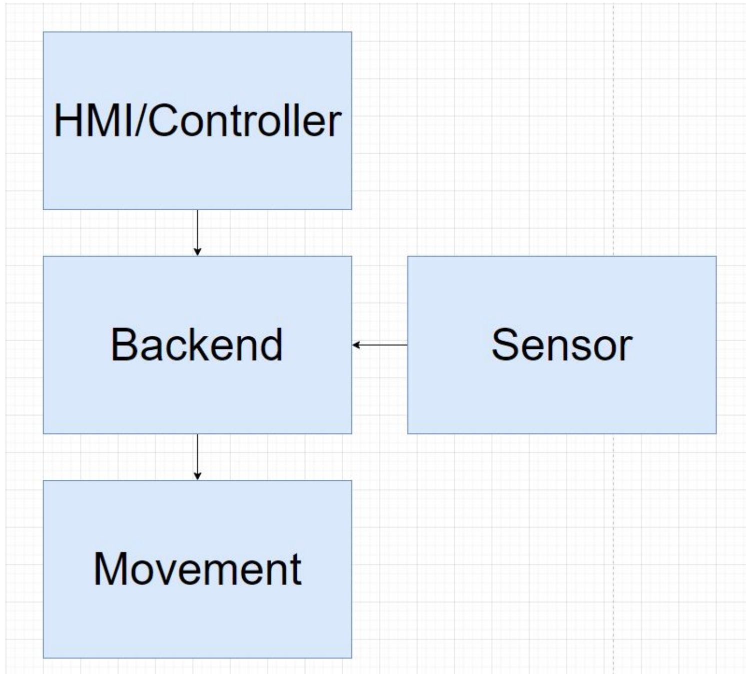



Figure 1: Robot Architecture Design Diagram

## 2.1 Human Machine Interface / Controller

The Human Machine Interface (HMI) allows the user to interact with the mechanical design and the functionalities of the underwater ROV. The HMI is a web based application that allows users to co-ordinate and control the movements of the ROV.The HMI sends outputs to the Movement System. Specifically for the movement system, the HMI will provide guidance to control the thrusters and ascend/descend. The HMI has separate interfaces that allow the user to control the depth level and the thruster activation. The HMI also provides a video feed to provide users guidance across the underwater environment.

## 2.2 Backend System

The Backend System will be used to host the two different HTTP servers. The first HTTP server will be used for taking requests from the HMI/Controller for movement controls. It will also send messages to the microcontroller through the serial port to control the individual hardware components for movement. The second HTTP server will be used for the pilot to see the camera footage being streamed from the robot.

## 2.3 Sensor System

The Sensor System is the system 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. In the sensor system, there will be a camera. The camera will stream video to the web app controller so the pilot can see what the robot sees.

## 2.4 Movement System

The Movement System consists of the parts involved for movement. For us, that will be the thrusters, and the ballast tanks subsystems. These subsystems will receive input from the Human Machine Interface / Controller that will control the power level on the thrusters and pumps regulating the movement of the robot. Specifically for the ballast tanks subsystem, the Movement System will receive input regarding how to control the pump responsible for filling the ballast tanks with water. Whether to fill the tanks or empty them. The ballast tank subsystem consists of the valves regulating the water flow and the pump that will fill the tanks with the water. The controller must be able to open and close the valves individually and turn on the pump.

# 3 Subsystem Definitions & Data Flow

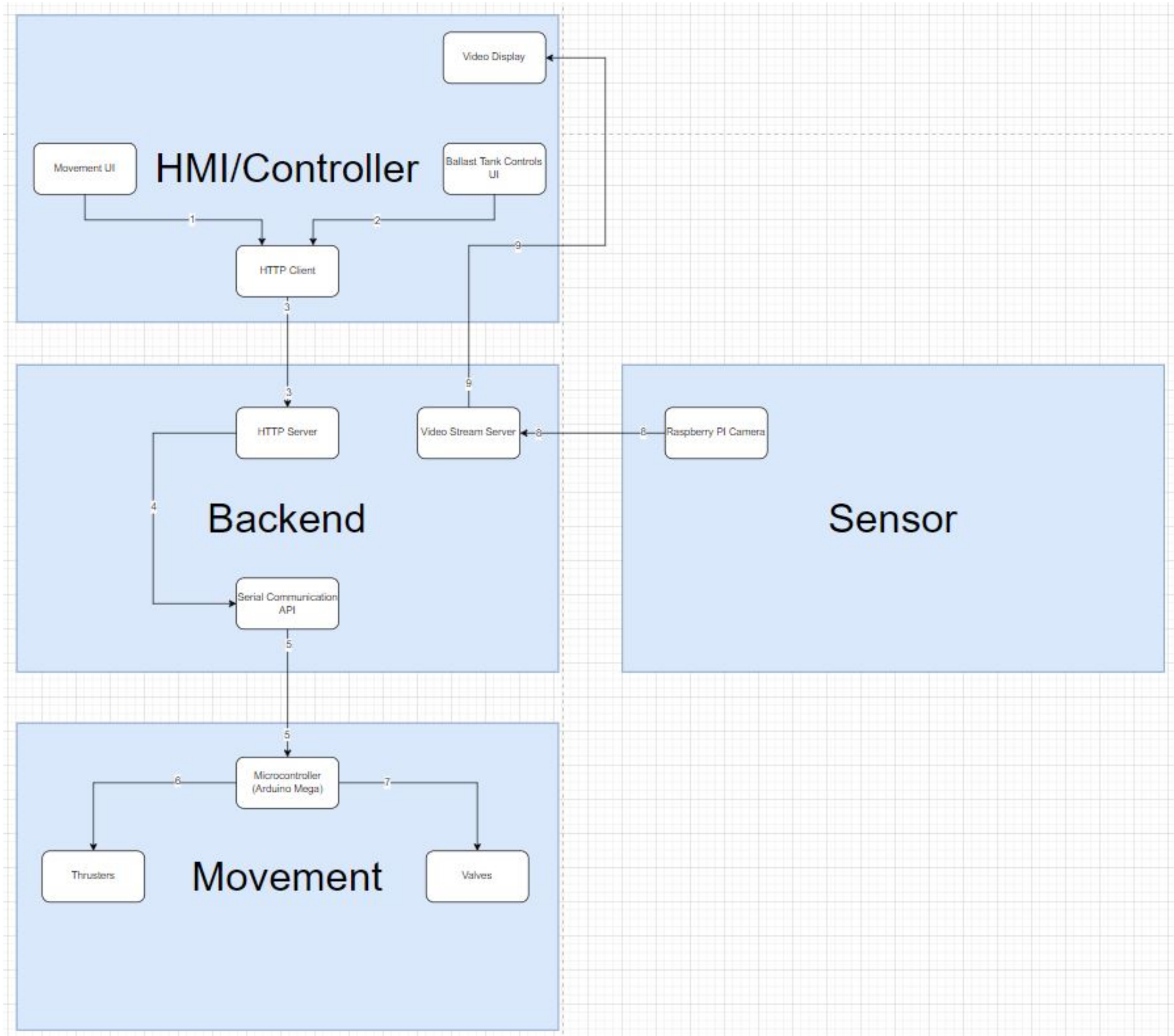



Figure 2: Robot Data Flow Chart

# 4 Human Machine Interface / Controller

The Human Machine Interface (HMI) allows the user to interact with the mechanical design and the functionalities of the underwater ROV. The HMI consists of four subsystems: Heads Up Display (HUD), Video Feed, Movement Controller, and the Depth controller.The Heads Up Display provides the user with the video feed from the robot. The video feed will allow the users to control and navigate movements of the robot with ease. The movement control interface will allow the user to control the thrusters which allow for movement in an xy-plane. The depth control interface will allow the user to control the fill levels on the individual ballast tanks and groups of the ballast tanks.

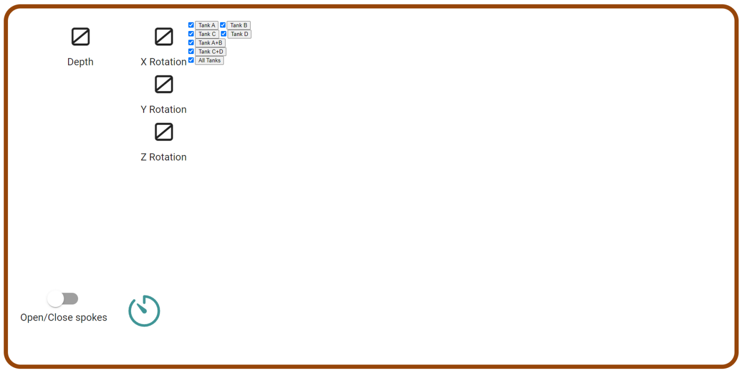



Figure 3: Design interface of the Controller without Video Feed

## 4.1 Video Display

The Video Display provides a live display of the underwater environment.

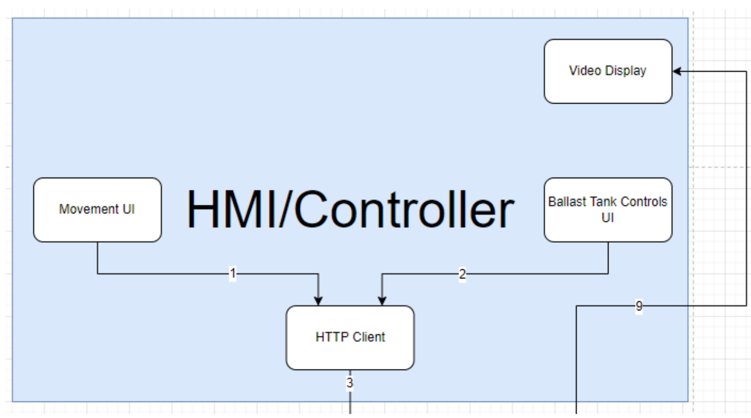



Figure 4: HMI Subsystem Diagram

### 4.1.1 Assumptions

The video display will only have a 180 degree field of view of the underwater environment with a 1080 p feed. An idea of 360 degree field of view was considered but the former was more feasible.

### 4.1.2 Responsibilities

The video display is responsible for providing the user with visual information that is vital for controlling the robot in an underwater environment. It will allow the user to be able to adjust the position and depth of the robot as obstacles come into view.

#### 4.1.3 Subsystem Interfaces

Table 2: Video Display Subsystem Interfaces

| ID | Description | Inputs | Outputs |
|---|---|---|---|
| #02 | Live feed of the environment | Camera Display | Video feed |

### 4.2 Movement UI

The Movement UI provides an interface to control the movement of the robot through the thrusters. It does this by picking up the key-presses of the user (w,a,s,d) to control the robot's thrusters. It will allow the robot to move forward, backwards, left and right. To turn right or left, only one of the two thrusters will be turned on to simulate angular movement.

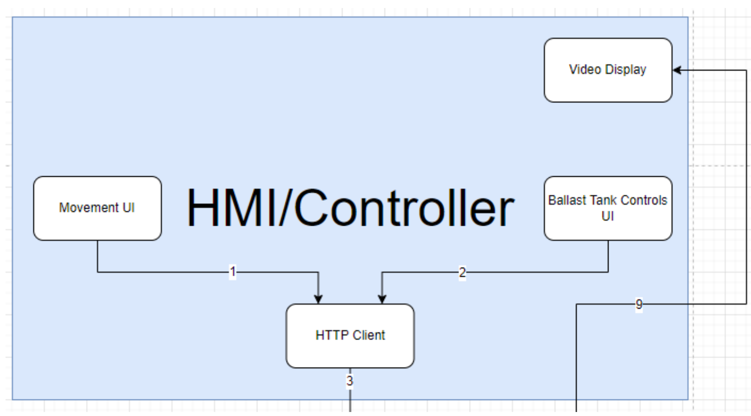



Figure 5: HMI Subsystem Diagram

#### 4.2.1 Assumptions

There will be two thrusters, one counterclockwise and one clockwise, that will be used. To turn right, the left thruster will be turned on, and to turn left, the right thruster will be turned on.

#### 4.2.2 Responsibilities

The movement controller provides a set of controls for forward, backward, left and right movement.

#### 4.2.3 Subsystem Interfaces

Table 3: Movement Controller Subsystem Interfaces

| ID | Description | Inputs | Outputs |
|---|---|---|---|
| #03 | Control the movement of the robot | User Inputs | Thruster Power |

### 4.3 Depth Controller

The depth controller allows the user to change the navigational depth of the robot. Specifically, it allows the robot to rise to a certain depth, descend to a certain depth and stay at a certain depth. This sub-system interacts with the Pump sub-system and Valve subsystem of the Movement System.

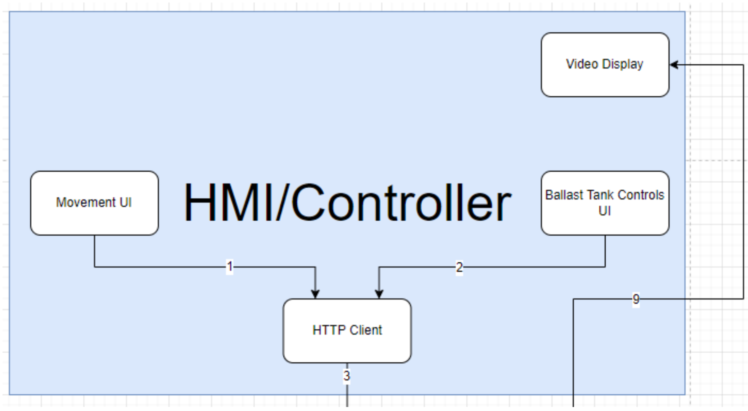



Figure 6: HMI Subsystem Diagram

#### 4.3.1 Assumptions

The depth controller interface is able to open/close the valves for the ballast tank on command. It also is able to control the pump subsystem (i.e determine how much water to push in or out) by adjusting the depth slider.

#### 4.3.2 Responsibilities

The depth controller is mainly responsible to change as well as maintain the desired depth required for perform any underwater task. It allows the user to control when the robot should rise to surface, go down to a certain depth and maintain the desired depth level.

#### 4.3.3 Subsystem Interfaces

Table 4: Depth Controller Subsystem Interfaces

| ID | Description | Inputs | Outputs |
|---|---|---|---|
| #04 | Control the system to empty or fill the ballast tanks | Checked<br>Empty | Valves for emptying<br>Valves for filling |
| #05 | Control the pumping mechanism for the ballast tanks | Ballast tank buttons | Ballast tanks targeted |

## 4.4 HTTP Client Component

The HTTP Client component allows the controller to send messages to the backend. It does this by sending HTTP Post requests to the HTTP server on the backend of the robot. These requests are composed of two fields, an action and the data for the action.

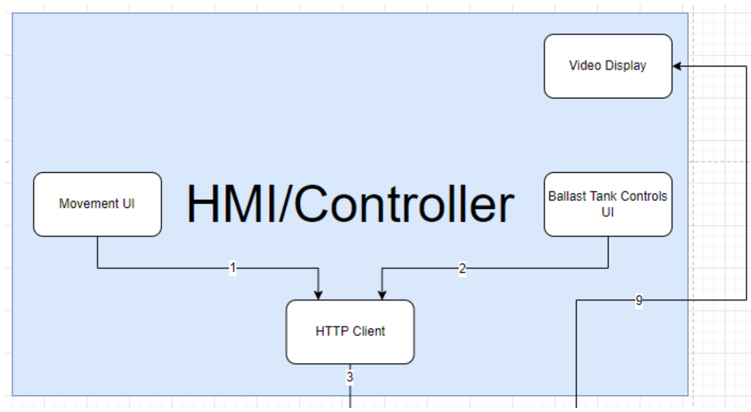



Figure 7: HMI Subsystem Diagram

### 4.4.1 Assumptions

The HTTP Post Request data field will be in JSON format.

### 4.4.2 Responsibilities

Responsible for communication between controller and the robot's backend.

### 4.4.3 Subsystem Interfaces

Table 5: HTTP Client Component Subsystem interfaces

| ID | Description | Inputs | Outputs |
|---|---|---|---|
| #06 | Control command | Action<br>Data for action | POST Request to Backend |

# 5 Sensor System

This layer of subsystems is the inputs that our drone will be able to capture so that we can effectively control it underwater. These readings will be to be to a certain level of accuracy and be communicated to the team at all points of the challenge. How effectually we complete this layer will be the deciding factor on if we are able to compete in the challenge.

## 5.1 Camera

Our video recording implement to see whats in front of the drone.

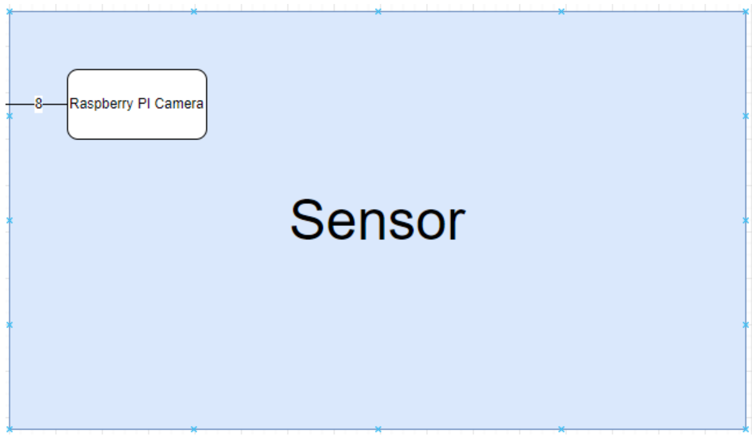



Figure 8: Sensor Subsystem Diagram

### 5.1.1 Assumptions

Able to work in underwater conditions or at least be able to be made protected from water. Conditions under the water shouldn't be to dark for a usable video feed.

### 5.1.2 Responsibilities

Need to be able to relay the area In front of the drone so those controlling it will be able to interact with course. Examples are the hoops that we will need to avoid, the block that we will need to pick up, and the button to release the tennis balls.

### 5.1.3 Subsystem Interfaces

Each of the inputs and outputs for the subsystem are defined here. Create a table with an entry for each labelled interface that connects to this subsystem. For each entry, describe any incoming and outgoing data elements will pass through this interface.

Table 6: Camera Subsystem Interfaces

| ID | Description | Inputs | Outputs |
|---|---|---|---|
| #01 | Video feed | N/A | Video Cable |
| #02 | Camera power | Power Cord | N/A |

# 6 Backend System

The Backend System consists of three main subsystems: an HTTP server for controls, an HTTP server for streaming video, and a serial communication API. The HTTP server for user-controls will receive the controls from the HTTP client component of the HMI/Controller. These controls will then be processed and used by the serial communication API to be sent over to the microcontroller in the Movement system. The control commands will be processed into a one byte message that contains the specified command for each action and data specified for that action. The HTTP server used for video streaming will allow the user/pilot to observe the field in front of the robot so the user can adjust the movement and depth of the robot. The serial communicatio API will be used to communicate with the microcontroller responsible for controlling the hardware systems.

## 6.1 HTTP Server

The HTTP server for user-controls will receive the controls from the HTTP client component of the HMI/Controller. The requests will all be POST requests and the data inside them will be used to construct the one byte command messages to be sent to the microcontroller.

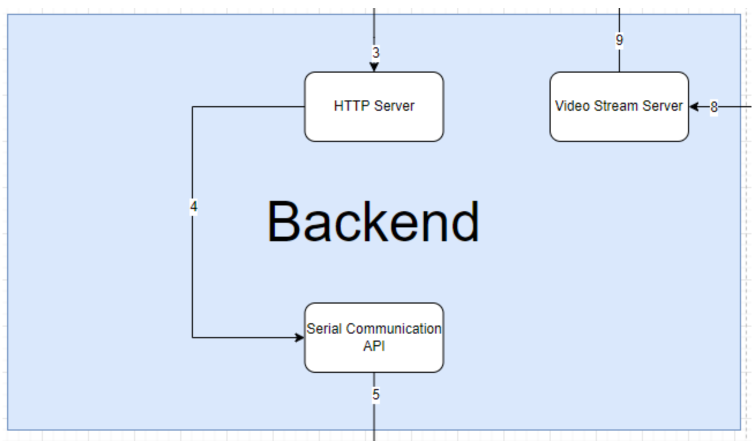



Figure 9: Backend Subsystem Diagram

### 6.1.1 Assumptions

All requests will be POST requests and the data field will be in JSON format.

### 6.1.2 Responsibilities

The HTTP server will be responsible for receiving controls from the controller.

### 6.1.3 Subsystem Interfaces

Table 7: HTTP Server Subsystem Interfaces

| ID | Description | Inputs | Outputs |
|---|---|---|---|
| #01 | HTTP Server | POST Request | Command |

## 6.2 Video Streaming Server

The HTTP Video streaming server will be used for streaming the video feed collected by the raspberry pi camera. This video feed will allow the user to pilot the robot with a first-person point-of-view.

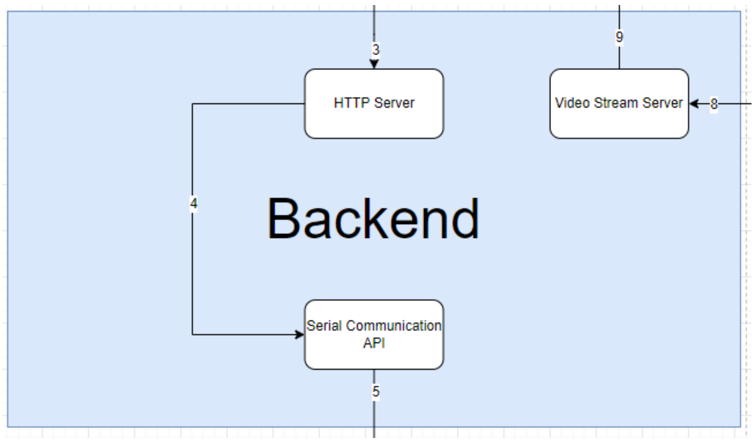



Figure 10: Backend Subsystem Diagram

### 6.2.1 Assumptions

The video feed will just images constantly updating.

### 6.2.2 Responsibilities

This HTTP server will be responsible for streaming video feed from the camera.

### 6.2.3 Subsystem Interfaces

Table 8: Video Streaming Server Subsystem Interfaces

| ID | Description | Inputs | Outputs |
|---|---|---|---|
| #01 | HTTP Video Server | Camera feed | Video stream |

## 6.3 Serial Communication API

The Serial Communication API will be used for processing the JSON data from the POST requests. It will also turn the data into one byte commands to be sent serially through the serial port to the arduino microcontroller.

### 6.3.1 Assumptions

There is an arduino at a specified port. The arduino is capable of serial communication.

### 6.3.2 Responsibilities

The serial communication API will be responsible for sending messages to the arduino through the serial port and processing the data into commands.

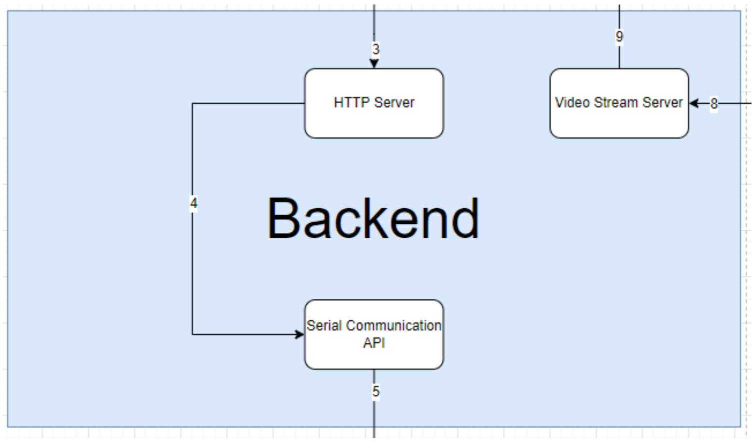



Figure 11: Backend Subsystem Diagram

### 6.3.3 Subsystem Interfaces

Table 9: Serial Communication API Subsystem Interfaces

| ID | Description | Inputs | Outputs |
|---|---|---|---|
| #01 | HTTP Video Server | Data from POST request | Byte command to microcontroller |

# 7 Movement System

The Movement System Layer consists of three main subsytems. The thrusters, the valves, and the pump. The thrusters will control horizontal movement (forward, backward, left, and right). The thrusters will be receiving input from the main control unit of the amount of power being fed to the thrusters. The valves will be controlling which tanks in the ballast tank system get filled. The input it receives is from the main control unit and whether or not to open. The pump is the final subsystem in this layer. It is responsible for filling the ballast tank system with water. It will receive input from the main control unit and how much to fill the tanks and for how long.

We considered using a piston-based ballast tank system originally. However, we felt that would increase the total weight of the system too much, so we opted in for the pump-based ballast tank system. In this, we traded off simplicity for overall weight of the system.

## 7.1 Propellors/Thrusters

The propellors/thrusters will be the part of the system used for horizontal movement.

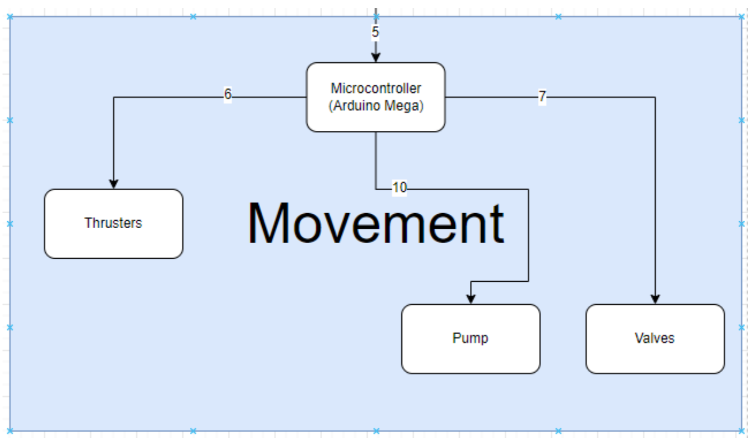



Figure 12: Movement Subsystem Diagram

### 7.1.1 Assumptions

To turn left or right, the opposite thruster in which direction we want to turn will be turned on. One thruster is counterclockwise and one is clockwise.

### 7.1.2 Responsibilities

The thrusters are responsible for moving the robot forward, backwards, left, and right. This subsystem will be responsible for the main movement of the robot.

### 7.1.3 Subsystem Interfaces

Table 10: Thrusters Subsystem Interfaces

| ID | Description | Inputs | Outputs |
|---|---|---|---|
| #01 | Power Cable to Thruster | Power Cord | N/A |

## 7.2 Valve Subsystem

The valves will be for controlling the flow of water into the ballast tank system (which tanks will be filled).

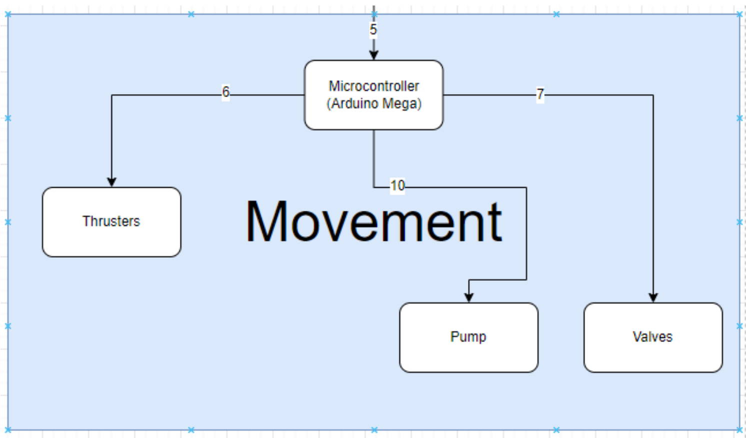



Figure 13: Movement Subsystem Diagram

### 7.2.1 Assumptions

Each set of two tanks (front and back tanks) will get a valve. These valves will be able to open and close independently of each other. Thus, allowing us to control the orientation of the robot under the water.

### 7.2.2 Responsibilities

The valves will be responsible for controlling the flow of water into the pairs of ballast tanks. They will be able to open and close independently of each other thus allowing the pairs of tanks to be at different water levels.

### 7.2.3 Subsystem Interfaces

Table 11: Valve Subsystem Interfaces

| ID | Description | Inputs | Outputs |
|---|---|---|---|
| #02 | Power Cable to Valve | Power Cord | N/A |

## 7.3 Pump Subsystem

The pump subsystem will be for filling the ballast tanks with water.

### 7.3.1 Assumptions

The pump is able to pump water both into the tanks and out of the tanks.

### 7.3.2 Responsibilities

The pump will be responsible for pumping water into the ballast tank system. It will control how much water gets pumped into or out of the system.

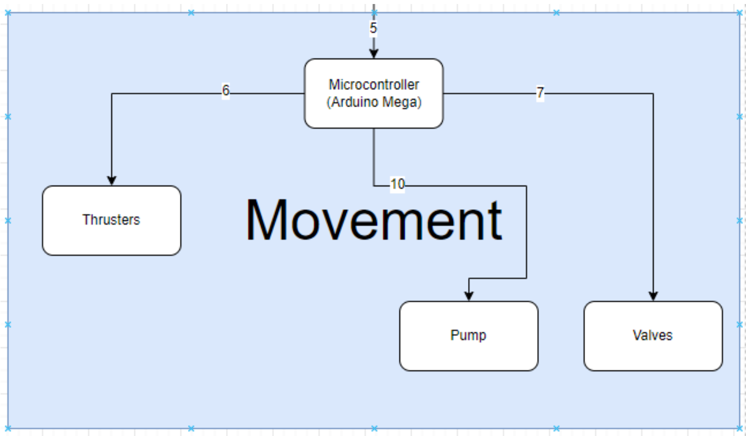



Figure 14: Movement Subsystem Diagram

#### 7.3.3 Subsystem Interfaces

Table 12: Pump Subsystem interfaces

| ID | Description | Inputs | Outputs |
|---|---|---|---|
| #03 | Power Cable to Pump | Power Cord | N/A |

### 7.4 Microcontroller Subsystem

The microcontroller will handle the processing of commands from the backend and using them to turn on the various hardware subsystems.

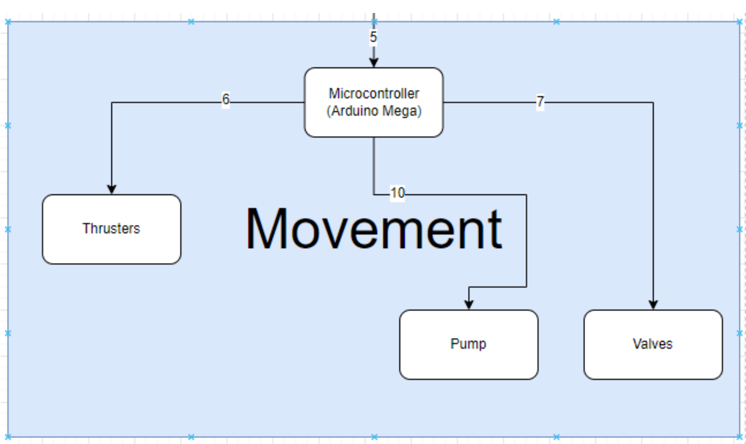



Figure 15: Movement Subsystem Diagram

#### 7.4.1 Assumptions

The microcontroller will receive inputs from the backend through the serial port.

### 7.4.2 Responsibilities

The microcontroller is responsible for controlling the various hardware systems used for controller movement of the robot.

### 7.4.3 Subsystem Interfaces

Table 13: Microcontroller Subsystem interfaces

| ID | Description | Inputs | Outputs |
|---|---|---|---|
| #03 | Serial Port | Byte command | Hardware controls |

## References