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

Architectural Design Specification
CSE 4316: Senior Design I
Fall 2021

TEAM: GAMMA

Team Gamma
IEEE Robotics Competition

Evan Cornish
Kartikey Sharan
Zachary Trumbaturi
Osama Siddiqui
Paola Gonzalez

Revision History

Revision	Date	Author(s)	Description
0.1	12.7.2021	EC, OS, ZT, KS, PG	document creation

CONTENTS

List of Figures

List of Tables

1 Introduction

This product will be made for the IEEE 2022 robotics competition in Houston Texas on April 2nd. The competition rules explains the full scope of the project but a short summary will be provided here. The objective of the competition is to make a underwater remote controlled submarine that is capable of collecting trash in hard to reach places. As such, the competition includes a number of obstacles that need to be navigated. The submarine will be remote controlled through a series of obstacles that involve hoops both forwards and backwards. Then the robot continues on to the second round. The objective will be to grasp debris that is meant to be mimicking trash in an ocean or river and navigate to a suppository station. The debris will be deposited and then more debris will be collected floating atop the water. That trash will then be deposited in a secondary trash station. In both rounds judges will be watching and award points based on speed, performance, and cost specifications of the robot. The faster and cheaper that the robot can operate the more points will be awarded.

2 System Overview

Our submarine design consists of 4 overall design systems. The control system, the movement system, the debris collection system and the visual system. The overall system works together delivering data between these various sub systems in order to insure that the pilot of the submarine can accurately execute actions related to operating the submarine.

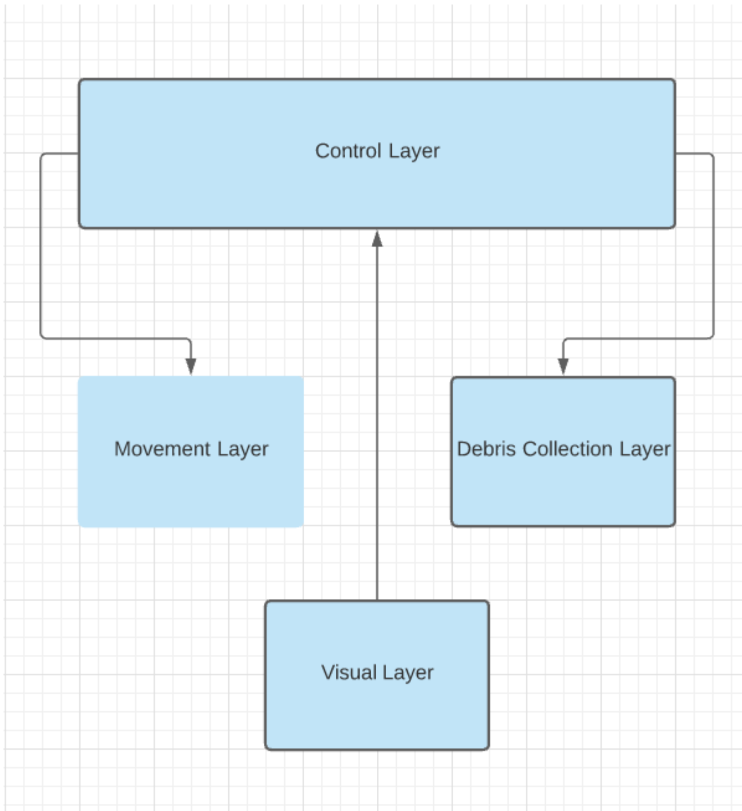

Figure 1: A simple architectural layer diagram

2.1 Control Layer Description

The control layer involves using the library ArduPilot to call a control system that will operate the underwater vehicle on 6 degrees of freedom. As such the vehicle will need the capability of turning on all axis as well as horizontal spins. It will send its commend through the tether connected to the vehicle and the motor operation will be updated according to the library specifications. The ArduSub sublibrary of the ArduPilot series of libraries has a specification with motor placement that must be followed in order to achieve full functionality from the library.

The control layer also receives input data through the tether from the visual system. That input data is displayed for the pilot onto a monitor system which contains an overlay with some targeting assistance graphics built on top of the display.

All of this data is manipulated through the tether which connects our generic RV controller with micro-controllers installed directly on the vehicle. The vehicle itself will contain between 2-4 micro-controllers that will be responsible for turning on and off the electrical signals.

2.2 Movement Layer Description

The movement layer currently consists of a set of 6 thrusters. The thrusters must be placed at predefined specified locations to conform to ArduPilot library's prebuilt specifications. In order to obtain 8 degrees of freedom we will need 2 rear thrusters, and then 4 thrusters placed at various points in a circular fashion all differing by 45 degrees from eachother. The thrusters are turned off and on via electrical signals from micro controllers installed on the vehicle in the control layer. The directional capability of the movement system is all handled automatically by ArduPilot, our system will only implement the thrusters in the specifications according to the ArduPilot library.

2.3 Debris Collection Layer Description

The debris collection layer consists of a mechanical arm to collect sunken debris, and a net pulley system to collect floating debris. The net pulley system will collect the floating debris and deposit the debris into a catapult system which will then launch the debris into the collection area. The mechanical arm system will receive signals to clasp the sunken debris from the control system and then release it via the same control system.

2.4 Visual Layer Description

The visual layer is the one layer that is meant to deliver information back to the control layer. The information is collected through a waterproof camera on the front of the submarine and then transmitted through the tether to be displayed on the monitor of the control system.

3 Subsystem Definitions & Data Flow

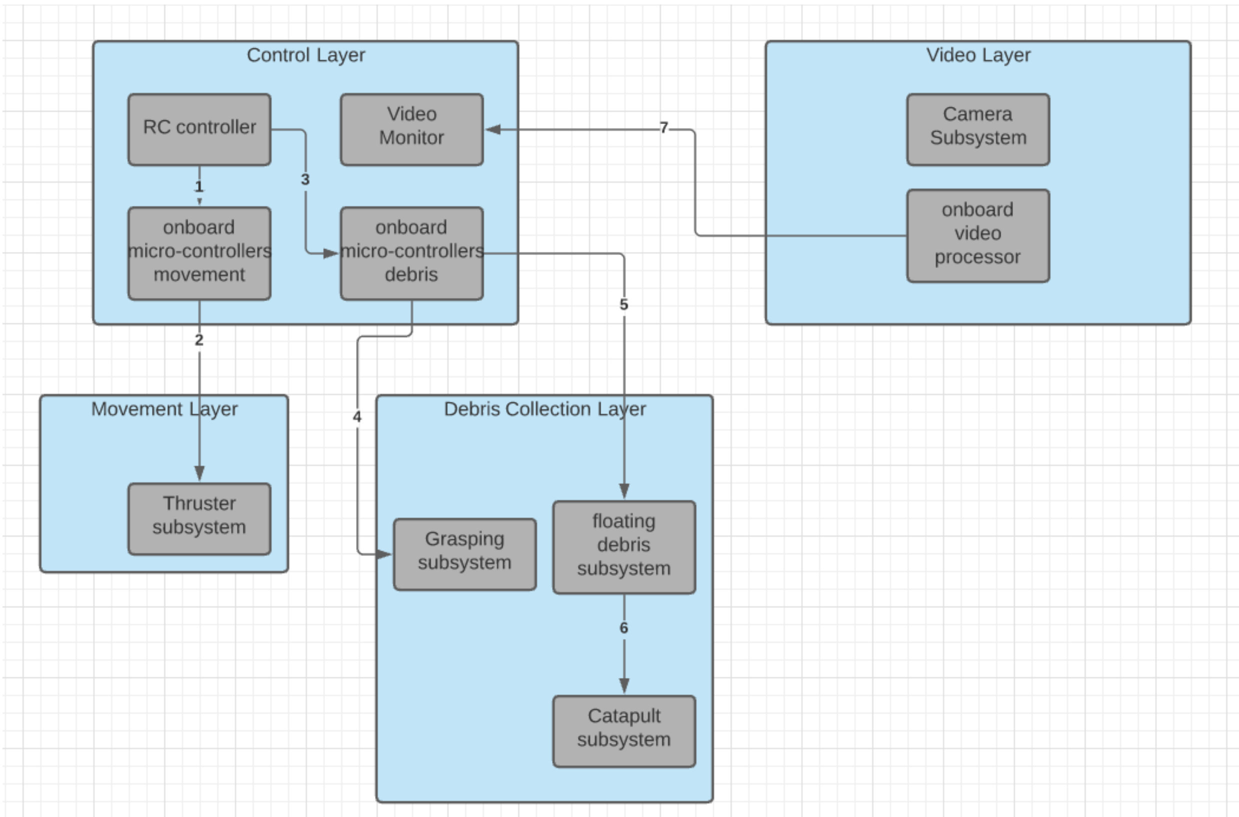

Figure 2: A simple data flow diagram

4 Control Layer Subsystems

4.1 Subsystem 1 RC controller

We've decided to use a double joystick RC controller for the time being. We have to spend the extra money to buy a remote controller but the benefit would be that the submarine will be easier to pilot. The other option was just using keyboard inputs from a laptop.

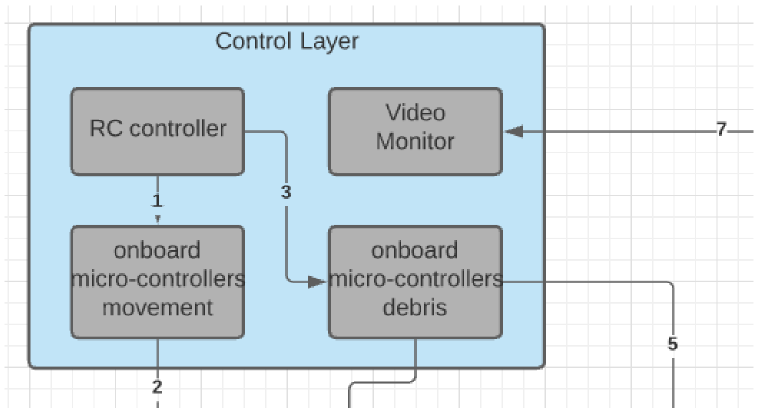

Figure 3: Example subsystem description diagram

4.1.1 Assumptions

We made the assumption that easier piloting will lead to higher points. We also assumed that an RC controller is going to be easier to use than keyboard inputs from a laptop.

4.1.2 Responsibilities

The responsibilities of the RC controller is to send the electrical signals to the laptop which will then be translated by ArduPilot into actionable instructions on the part of the micro-controllers.

4.1.3 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 2: Subsystem interfaces

ID	Description	Inputs	Outputs
#01	USB interface through tether	joystick forward backward joystick left-right	ArduPilot library calls
#02	USB interface	button clasp	micro-controller signal

4.2 Subsystem 2 onboard micro-controllers movement

4.2.1 Assumptions

We've made the assumption that all movement can be controlled by turning power on/off to the thrusters.

4.2.2 Responsibilities

The responsibility is for the micro-controller to give power and/or cut power to the thrusters depending on how we want the submarine to move. The signals for which action the microcontrollers will take will be given via translated library calls from ArduPilot.

4.2.3 Subsystem Interfaces

Table 3: Subsystem interfaces

ID	Description	Inputs	Outputs
#	USB interface through tether	USB input	Arduino pin signals

4.3 Subsystem 3 onboard micro-controllers debris

4.3.1 Assumptions

we've made the assumption that we will not be able to figure out how to get the arm to clasp without some sort of electrical power solution.

4.3.2 Responsibilities

The responsibility of this subsystem is to force the clamp to close when we have the target in sight, and to launch the catapult debris system when it is loaded.

4.3.3 Subsystem Interfaces

Table 4: Subsystem interfaces

ID	Description	Inputs	Outputs
#	USB interface through tether	USB input	Arduino pin signals

4.4 Subsystem 4 video monitor

4.4.1 Assumptions

we've made the assumption that it will be impossible to pilot the submarine by looking into the water at an angle due to the light reflectivity.

4.4.2 Responsibilities

The responsibility of this subsystem is to display the video information being displayed to it from the camera system.

4.4.3 SUBSYSTEM INTERFACES

Table 5: Subsystem interfaces

ID	Description	Inputs	Outputs
#	USB interface through tether	USB input	video image

5 Movement Layer Subsystems

5.1 Subsystem 1 thruster subsystem

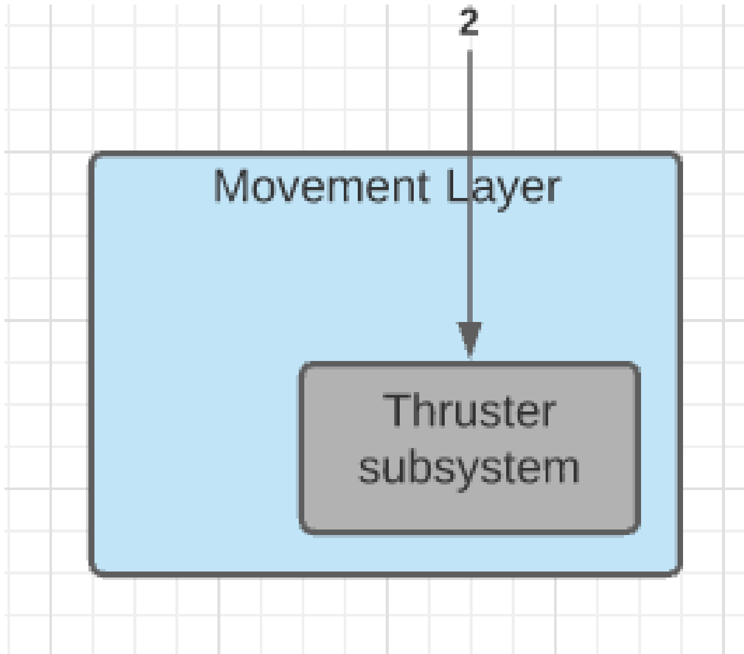

Figure 4: Example subsystem description diagram

5.1.1 Assumptions

We made the assumption that placement of thrusters and general thruster design would be better to copy than to design on our own. We're using our design specifications in accordance with the ArduPilot library documentation.

5.1.2 Responsibilities

the responsibility of the thruster subsystem is to move the robot by pushing water through the thrusters. It can also reposition the robot in accordance to the ArduPilot specifications with 8 degrees of freedom.

5.1.3 Subsystem Interfaces

Table 6: Subsystem interfaces

ID	Description	Inputs	Outputs
#xx	copper wire interface	electricity	kinetic motion

6 Debris Layer Subsystems

6.1 Subsystem 1 grasping subsystem

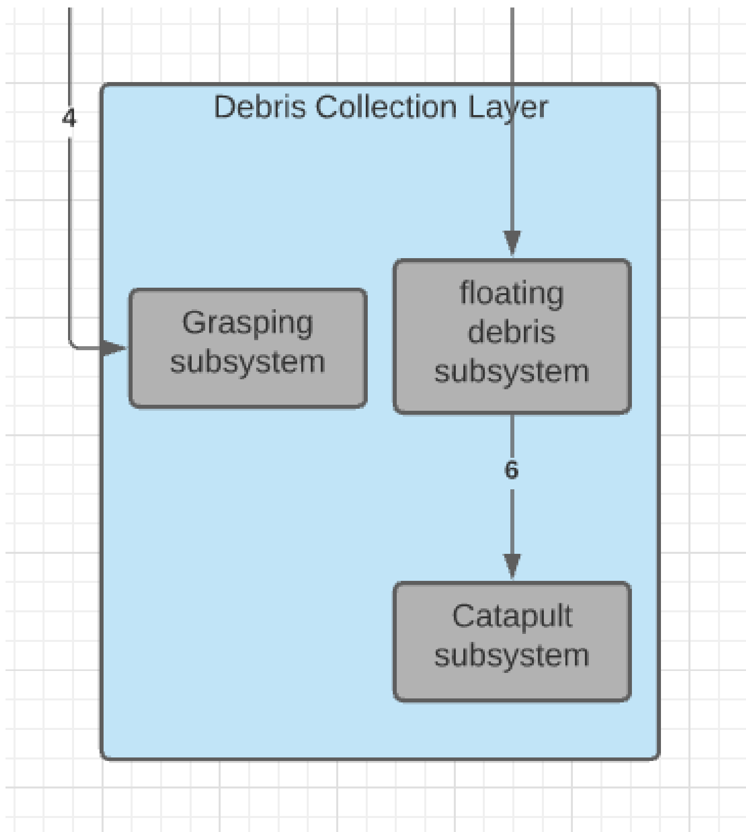

Figure 5: Example subsystem description diagram

6.1.1 Assumptions

Since the debris collection is clearly spelled out in the competition instructions, no assumptions are necessary for this subsystem.

6.1.2 Responsibilities

Grab the underwater debris, be able to hold onto the debris while the submarine is navigating through water with water resistance pushing the block, and then release the debris in the deposit location.

6.1.3 Subsystem Interfaces

Table 7: Subsystem interfaces

ID	Description	Inputs	Outputs
#xx	Description of the interface/bus	input 1 input 2	output 1
#xx	Description of the interface/bus	N/A	output 1

6.2 Subsystem 2 floating debris subsystem

6.2.1 Assumptions

Since the debris collection is clearly spelled out in the competition instructions, no assumptions are necessary for this subsystem.

6.2.2 Responsibilities

collect debris floating on top of the water and deposit it in the catapult system. This will be done kinetically so there are no inputs to this mechanism.

6.2.3 Subsystem Interfaces

Table 8: Subsystem interfaces

ID	Description	Inputs	Outputs
#xx	Description of the interface/bus	input 1 input 2	output 1
#xx	Description of the interface/bus	N/A	output 1

6.3 Subsystem 3

6.4 Subsystem 2 catapult subsystem

6.4.1 Assumptions

We assumed it will be necessary to launch the debris up a little bit in order to get it over the lip of the deposit station.

6.4.2 Responsibilities

launch the floating debris that we collected into the deposit station.

6.4.3 Subsystem Interfaces

Table 9: Subsystem interfaces

ID	Description	Inputs	Outputs
#xx	Description of the interface/bus	input 1 input 2	output 1
#xx	Description of the interface/bus	N/A	output 1

7 Video Layer Subsystems

7.1 Subsystem 1 camera subsystem

This section should be a general description of a particular subsystem for the given layer. For most subsystems, an extract of the architectural block diagram with data flows is useful. This should consist of the subsystem being described and those subsystems with which it communicates.

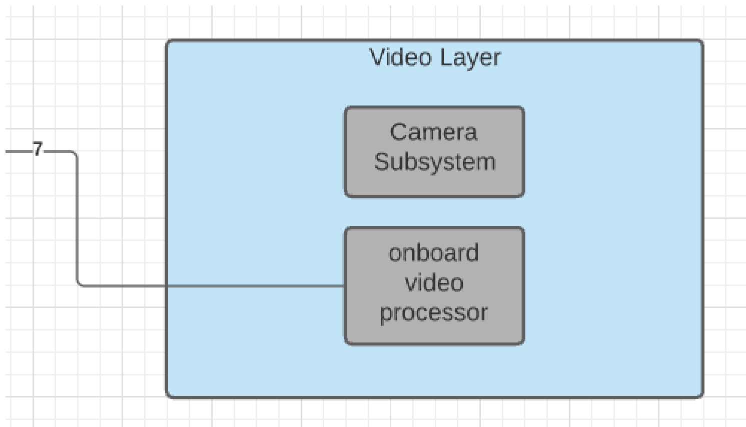

Figure 6: Example subsystem description diagram

7.1.1 Assumptions

We made the assumption that we would not be able to navigate the submarine without an underwater perspective.

7.1.2 Responsibilities

the responsibility of the camera is to capture the perspective of the submarine and relay it back to the video monitor.

7.1.3 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 10: Subsystem interfaces

ID	Description	Inputs	Outputs
#xx	USB interface	USB	USB port to Raspberri pi

7.2 Subsystem 2 onbaord video processor

7.2.1 Assumptions

We made the assumption that we the video will need to be relayed to some sort of small computer for processing and then sending on its way to our video monitor system.

7.2.2 RESPONSIBILITIES

take the live USB feed of the video camera and propagate it back to our video monitor via the tether.

7.2.3 SUBSYSTEM INTERFACES

Table 11: Subsystem interfaces

ID	Description	Inputs	Outputs
#xx	USB interface	USB	Tether wire

REFERENCES