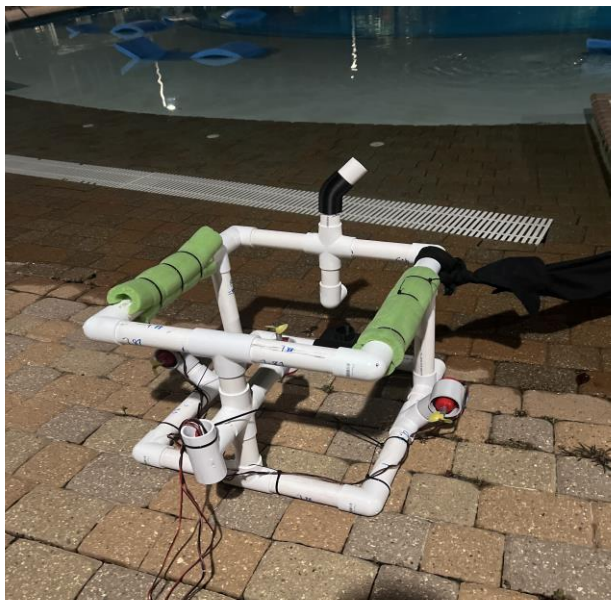
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

JellyBot

Submitted to:

Timothy Sellers
ECE 4542: Senior Design II
Department of Electrical and Computer Engineering
413 Hardy Road, Box 9571
Mississippi State University
Mississippi State, Mississippi 39762



November 8, 2024
Prepared by:

Buhlebenkosi Maposa, Jaden Hall, Jennefer Jones, Tyler Lane, Roland Robinson
Faculty Advisor: Professor Timothy Sellers
Department of Electrical and Computer Engineering
Mississippi State University
413 Hardy Road, Box 9571
Mississippi State, Mississippi 39762
email: {bm2330, jmh1543, jpj183, jtl418, rfr63 }@msstate.edu

LIST OF ABBREVIATIONS

ABS – Acrylonitrtile Butadiene Styrene

CAD – Computer Aided Design

DC – Direct Current

IEEE – Institute of Electrical and Electronics Engineers

In - Inch

I/O - Input/Output

Lbs. - Pounds

LED – Light Emitting Diode

Li-Ion – Lithium-Ion

mAh – Milliampere-hour

OS – Operating System

PCB – Printed Circuit Board

PLA – Polyactic Acid

PVC – Polyvinyl chloride

+

Qi – Wireless Charging Method

ROV – Remotely Operated Vehicle

UART – Universal Asynchronous Receiver-Transmitter

USB – Universal Serial Bus

USB-C – Universal Serial Bus Type C

V – Voltage/Volts

VAC - Voltage Alternating Current

VCD –Voltage Direct Current

EXECUTIVE SUMMARY

Pollutants are an enduring complication in most aquatic ecosystems. The occurrence of these pollutants, especially plastics, is difficult to mitigate due to the abundance of disposable, non-biodegradable products that humans use daily. Many of these pollutants reside underneath the water's surface and are difficult to remove from the environment. While products exist for trash collection from aquatic environments, these products tend to lack the ability for underwater retrieval. Other options that would allow for underwater retrieval require a person to be in the water themselves creating a possible detrimental effect on both the ecosystem and the person. Many freshwater ecosystems are either cleaned by the public, environmental workers, or government agencies which can lead to human injuries as well as a lack of time for other tasks.

The JellyBot is a freshwater ROV that has an arm that is attached to it to collect underwater trash. The debris it was made to collect can range from the size of a grocery bag to the size of a soda bottle. It is entirely able to be controlled remotely allowing for less human intervention. A user would no longer have to go into the water to make sure that underwater debris is collected. This would protect the freshwater ecosystem from possible damage due to human intervention and protect the user from possibly being attacked by fauna or sustaining injuries in other ways. The robot is entirely waterproof to allow it to work underwater. The arm is extended outward from the robot to allow the user to be able to control the arm without worrying about hitting the frame of the robot leading to any damage. The lights are extended outward as well to allow for the camera to pick up anything that is below or in front of the robot. The camera is extended as well to allow for a well-lit visual. The camera is also connected to a livestream on a user's device so that a user can see, allowing them to stay out of the water and avoid potential damage to the robot from the environment.

The design of the JellyBot was heavily influenced by the constraints set forth in its conceptualization. The initial constraints were economically based, focusing on the cost of the robot and how much time it would take to complete. To meet the cost constraint, many parts were ordered using the budget as well as outside of the budget. In addition, to meet the time constraint, the jellybot's design was simplified to look like a standard homemade underwater ROV rather than another physical design. The next constraint was environmental. For the JellyBot to be used underwater, it had to be fully waterproof, which was achieved using silicone and rubber sealant on all parts of the robot. This also helped with the health and safety constraint, since the silicone sealant kept all electronics from interacting with the water around them. Many of the electronics are also placed on land with long tethers to connect the JellyBot. The next three constraints related to remote usage of the JellyBot. The JellyBot is controlled using a remote with the receiver being connected to the robot via a wired tether. The tether both acts as a retrieval device as well as a communication device between the remote and the robot. Lastly, the buoyancy constraint was controlled by creating a system where the robot pushes water above it to sink while its idle state allows it to float just below the surface of the water.

Our design is unique because although underwater ROVs exist with a similar framework, they do not have the capabilities of underwater trash retrieval made possible through the integration of a camera, lights, and an arm. Both the movement of the robot itself and the movement of the robotic trash collection claw is user-friendly through use of remote control. The camera also sends data straight back to the user in real time. Planned improvements to the design include modifying its size, redesigning the robotic arm system, or adding a collection bin housed on the robot so that the user does not have to collect one piece of trash at a time.

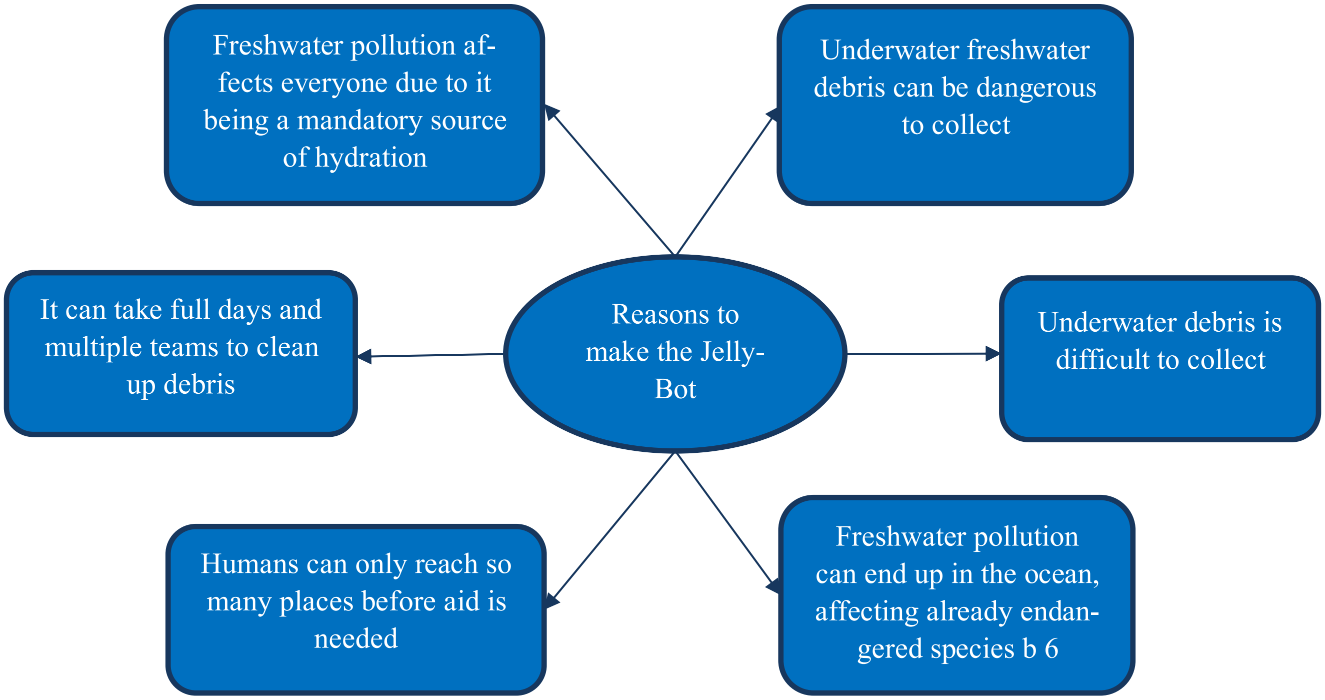


Figure 1. Graphic of reasons for making the JellyBot

TABLE OF CONTENTS

1. DESIGN REQUIREMENTS/CONSTRAINTS

Underwater pollution in freshwater affects both freshwater and saltwater ecosystems. By cleaning the debris in freshwater environments, the number of pollutants that reach saltwater environments is minimized. The JellyBot was developed and designed to clean underwater freshwater debris such as plastic bottles and bags in order to aid human efforts in freshwater conservation. Below are the constraints that needed to be met so that the JellyBot could do exactly what it was designed to do in the environment. They are divided into technical design constraints, practical design constraints, and engineering standards.

1.1 **Technical Design Constraints**

The Technical Design Constraints shown in Table 1.1 describe JellyBOT's minimum requirements to properly function.

Table 1.1. Technical Design Constraints

Name	Description
Buoyancy and Stability	The JellyBot achieves neutral buoyancy (tolerance ±10 grams) with materials and design that allow for stable, controlled movement underwater.
Waterproofing and Electrical Insulation	Waterproofing withstands depths of up to 3 meters (10 feet) for at least 30 minutes without any leakage.
Propulsion and Maneuverability	Thrusters provides a minimum thrust-to-weight ratio of 1:1, allowing JellyBot to overcome its weight in water and maneuver effectively.
Power Supply and Battery Life	All battery capacities allow for a minimum operational duration of 1 hour on a single charge. This will depend on individual power consumption.
Debris Collection Mechanism	The collection mechanism has a capacity to hold up a 16-ounce bottle without affecting the JellyBot buoyancy excessively

In the below sections, the technical constraints are expanded upon.

1.1.1 **Buoyancy and Stability**

Firstly, for the robot to be used underwater buoyancy, stability, propulsion and maneuverability were important constraints to consider. If any of the four were neglected, the likelihood of JellyBot working properly would be slim to none. For buoyancy and stability, we chose to have a neutral buoyancy of ±10 grams because our robot is naturally buoyant because of its structure being made of PVC. It is a reminder to constantly modify what we have as we added pieces to the JellyBot. It is precise enough to ensure that the JellyBot will not continuously sink or float while allowing forsmall changes in weight. It also helps the robot maintain a more stable depth without requiring constant input from a potential operator to correct its position.

1.1.2 Waterproofing and Electrical Insulation

The waterproofing and electrical insulation must ensure that no water will seep into any of the electrical components of the JellyBOT. The waterproofing must also withstand water pressures that occur in depths of up to 10 feet. All components that encounter water are insulated with various waterproofing materials including silicon caulk and rubber. The camera, lights, and any wiring that is in contact with the water is fully waterproofed to avoid the shorting of any circuitry, which would be detrimental to the functionality of the robot.

1.1.3 Propulsion and Maneuverability

For propulsion and maneuverability, the thrusters provide a minimum thrust to weight ratio of 1:1 so that it can efficiently maneuver through an underwater environment. This ratio allows for the robot to move without overshooting targets. There are two thrusters that control horizontal motion and one thruster that controls vertical motion. They are strategically placed so that the robot is well-balanced and resists unwanted rotation in the wrong direction. The thruster propellers are also capable of spinning clockwise and counterclockwise so that the robot can move forwards and backwards as well as up and down.

1.1.4 Power Supply and Battery Life

The JellyBOT's battery systems are designed to support a minimum operational duration of one hour per charge, ensuring sufficient time for effective underwater tasks. However, actual runtime may vary based on the specific power consumption demands of each component during operation.

1.1.5 Debris Collection Mechanism

The collection mechanism is designed to hold up to a 16-ounce bottle without significantly impacting JellyBot's buoyancy. This capacity allows the robot to retrieve waste effectively while maintaining stable, neutral buoyancy for smooth operation underwater. Trash that is too heavy or large can affect the weight distribution of the robot, causing issues in the steerability of the JellyBOT.

1.2. Practical Design Constraints

This section highlights JellyBOT's practical design constraints, based on an ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors.

Table 1.2. Practical Design Constraints

Type	Name	Description
Economic	Cost	The design team has a budget of $1,000 to build the Jelly-BOT.
Health and Safety	Public Health	Considering how the robot's operation could impact local ecosystems and water quality, ensuring it does not introduce pollutants is highly important.
Environmental	Environmental Impact	Designing the robot to minimize its environmental footprint, using eco-friendly materials and protecting aquatic ecosystems.

Social	Social Factor	Designing the robot to be user-friendly and adaptable to various environmental conditions, contributing to global environmental efforts.
Communication	Communicaton	Communication to all components' receiving signal is possible.

1.2.1 Cost

The Jelly-BOT is designed within a $1,000 budget. This limit impacts material and component choices, requiring cost-effective alternatives while still meeting performance standards. The design team must balance affordability with functionality, ensuring the robot performs effectively without exceeding the budget.

1.2.2 Public Health

The Jelly-BOT must operate without harming local ecosystems or water quality. The robot is built with non-toxic materials and waterproof components to prevent pollution and ensure safe operation in aquatic environments, safeguarding public health and the environment.

1.2.3 Environmental Impact

The Jelly-BOT is designed to minimize its environmental impact. Eco-friendly materials are used where possible, and the robot is built to be energy-efficient and durable. This reduces waste and ensures that minimal harm comes to the aquatic ecosystems while performing its cleaning tasks.

1.2.4 Social Factor

The Jelly-BOT is user-friendly and adaptable, making it accessible to a wide audience. Its simple interface allows people from various backgrounds to use it effectively, contributing to global environmental efforts such as cleaning freshwater ecosystems.

1.2.5 Communication

Reliable communication is crucial for Jelly-BOT's operation. Wireless signals control the movement of the robotic arm and the camera's visuals, ensuring smooth interaction between the user and the robot. Effective communication allows for real-time control and error-free operation, especially in underwater environments.

1.3 Engineering Standards

Table 1.3 highlights the appropriate engineering standards and guidelines followed during the process of making and using the JellyBOT.

Table 1.3. Appropriate Engineering Standards

Standard	Standard Document	Specification
Ingress Protection X8	International Electrotechnical Commission Standard 60529	Immersion (1 meter or deeper) [8]
Institute of Electrical and Electronics Engineers 45.4	IEEE Recommended Practice for Electrical Installations on Shipboard-Marine Sectors and Mission Systems	Electrical safety practices followed to prevent shocks. Grounding is required [9] Minimize interference with other systems for reliable communications [9] To avoid damage, use materials that resist corrosion. This protects electrical components from water exposure [9]
Federal Communications Commission (FCC) Part 15	Code of Federal Regulations Title 47, FCC, Part 15	Does not cause interference and can accept any inference that may be received [10]

1.3.1 Ingress Protection (IPX8)

This rating under the IEC Standard 60529 specifies a high level of water resistance. An IPX8 rating means the equipment is suitable for continuous immersion in water at depths of 1 meter or more, which is essential for underwater robotics.

1.3.2 Institute of Electrical and Electronics Engineers (IEEE) 45.4

This standard provides guidelines for safe and effective electrical installations on marine vessels. It emphasizes practices to prevent electric shock through grounding and promotes reliable communication by minimizing interference with other systems. Additionally, it recommends using corrosion-resistant materials to protect components from saltwater damage. Though the JellyBot is made for freshwater usage, this standard sets a clear guideline for underwater vessels as a collective.

1.3.3 Federal Communications Commission (FCC) Part 15

This FCC regulation (47 CFR, Part 15) mandates that electronic devices must not emit harmful interference and must accept any interference received. This standard is critical for maintaining clear, reliable communication in systems like remote-controlled robotics in shared frequency environments.

Following these industry standards allows Jelly-BOT to withstand freshwater environments and electrical interference while communicating with the remote, camera, and any other devices either on or connecting to the robot.

2. APPROACH

The JellyBOT is an underwater robot designed for easy and efficient trash retrieval. Constructed with a PVC frame for neutral buoyancy, it is equipped with three propellers that enable smooth left, right, and vertical movement, all controlled remotely. Integrated lights illuminate the surroundings, allowing the operator to use the onboard camera to locate and identify trash for collection with a robotic arm. The battery charges wirelessly via electromagnetic induction, making JellyBOT a fully integrated, effective tool for underwater cleanup.

2.1. Hardware

JellyBot's hardware consists of five key subsystems that work together to enable efficient underwater trash retrieval. The remote-control subsystem manages the robot's movement and depth through a dedicated remote control. The collection subsystem controls the arm and claw, which are operated by servos that allow for precise arm movement on a five-point axis. A separate remote controls the arm's movement as well as the opening and closing of the claw. The power subsystem features a high-capacity battery and a wireless charging system, ensuring long operation times and easy recharging without compromising water safety. The visual subsystem includes a camera to aid in debris identification and visualizing the environment, along with integrated lights to illuminate the surroundings for better visibility. The propulsion subsystem consists of motors and propellers for movement in water. These components are integrated to work seamlessly together, allowing JellyBOT to move, collect debris, and operate efficiently in harsh underwater environments.

2.1.1. Remote-Control Subsystem

The remote-control subsystem for JellyBot needed to meet specific criteria, including reliable communication range, efficient power usage, and sufficient transmitter channels for precise seamless movement control. Cost effectiveness played a crucial role, as the team needed a budget-friendly solution without sacrificing functionality. Weight considerations were also made as it was critical to maintaining the robot's buoyancy and overall performance underwater. After evaluating available options, the design team chose a remote that balanced these requirements, ensuring reliable and effective operation in underwater conditions.

Table 2.2: Remote Control Options

Model	Range of Communication	Power Intake	Weight	Transmitter Channels	Cost
Requirements	Minimum of 12 m	≤ 5 volts	< 2.3 kgs.	> 5 channels	< $80

Flysky FS-i6 Model 2 [3]	500 m	< 4.2 V	< 0.9 kgs.	6 channels	$43.75
Flysky FS-i4 [4]	~	< 4.2 V	< 0.9 kgs.	4 channels	$37.99
Radiolink AT10ll [5]	3862 m	7.4 - 15.0 V	1 kg.	12 channels	$158.89
BETAFPV ELRS LiteRa-dio 3 [6]	~	~	1 lb.	8 channels	$72.99

The Flysky FS-i6 remote control was chosen for JellyBot due to its ability to meet the project's requirements while staying within budget constraints. The Flysky FS-i6 is equipped with a range of customizable controls, including joysticks, buttons, and switches, all programmed to precisely manage JellyBot's articulation and movement. Additionally, the remote control includes a transmitter, which is essential for sending commands to the robot. This transmitter supports six channels, enabling the remote to transmit up to six separate signals. This capability allows the team to issue up to five distinct commands to JellyBot, facilitating detailed control over its movements and actions.

2.1.2. Camera and Lighting Subsystem

For JellyBot's camera integration, key requirements included high image clarity, compatibility with the Raspberry Pi, and seamless connectivity to a laptop, desktop or other device for live monitoring. The camera needed to be robust enough for underwater use while providing sufficient brightness for visibility in low-light conditions. Additionally, the camera had to allow for streaming capabilities to create real-time visuals of the surroundings of the robot and the debris being collected. These specifications ensure reliable operation and precise control in underwater environments.

Table 2.3: Camera Options

Model	**Compatibility**	**Integration Simplicity**	**Brightness**	**Image Clarity**	**Price Per Camera**	**Total Price**
Require-ments	Compatibility with Raspberry PI	Need for any additional Parts	Affected by underwater use	Affected by under-water use	> $30	< $60
MUNFEE Underwater Ice Fishing Camera	Yes	Yes, an exter-nal tether sys-tem	Yes	Yes	$22.19	$133.14
Notebook In-ternal Webcam	Yes	No	Moderately	Moder-ately	$9.18	$55.08
Raspberry Pi Camera Mod-ule	Yes	No	No	No	9.99	$9.99

The Raspberry Pi camera module was selected for JellyBot due to its ability to meet the project's imaging requirements while staying within budget constraints. The camera provides clear, high-quality images essential for the robot's operations, including navigation and environmental monitoring. It is compatible with both the Raspberry Pi and Arduino Nano V3, ensuring smooth integration into the system without the need for an external module. It also has streaming capabilities which can be easily initiated through lines of code and knowledge of the raspberry pi's IP address which can be easily found. The Raspberry Pi camera module is easy to implement, offering reliable performance without the complexity of Linux-based code or extended wiring required by other camera options, making it an ideal choice for the team's needs.

2.1.3 Propulsion Subsystem

Table 3-6 highlights the various options considered for the thruster used in JellyBOT based on factors such as voltage levels, horsepower, size, and price.

Table 2.4: Thruster Options

Model	**Voltage**	**Horsepower**	**Size**	**Total price**
Requirements	< 30	<1	Equilateral dimensions	< 50
XHSESA 12V-24V 20A Underwater Thruster Motor [25]	12-24 V	½ HP	No	$29.98
Submersible Boat Bilge Water Pump 12v 1100gph Non-Automatic Marine Electric Bilge Pump 390 W [26]	12 V	1/8 HP	Yes	$14.99
DNYSYSJ Brushless Motor Underwater Thruster Motor [27]	12-24 V	1 HP	Yes	$21.00

The chosen thruster is the 1100 GPH Bilge Pump. A 1100 GPH bilge pump is an economical choice for an underwater ROV at a price of $14.99, offering a reliable balance of power and affordability. With a typical operating voltage of 12V, it's compatible with standard ROV battery configurations, ensuring a steady power supply. The motor's horsepower rating is often around 1/8 HP, which provides sufficient thrust to maneuver the ROV efficiently without draining the battery too quickly. This combination of high flow rate, low voltage requirement, and affordable cost makes the 1100 GPH bilge pump a practical option for propulsion in underwater robotics and for the design team.

2.1.4 Actuator Subsystem

Table 2.5: Servo Motors

Model	Voltage	Torque	Price	Degrees
Miuzei 20KG Servo Motor High Torque RC Servo	6.8V	20kg	$13.59	270
Miuzei MG90S 9G Micro Servo	6V	9G	$13.99	180

The actuator subsystem of the jellybot combines larger high-torque servos and smaller micro servos to enable versatile motion control. The larger servos support the base of the robot's arm, providing the strength required to lift and carry objects. Meanwhile, the micro servos handle precise movements, such as closing the gripper or adjusting its orientation. This combination ensures the jellybot achieves both robust load handling and fine motor control for delicate tasks.

Table 2.6: Microcontroller

Model	Voltage Input	Voltage Output	Price
Freenove Breakout Board for ESP32	7-12V	3.3V - 5V	$12.95
Freenove ESP32-WROOM	3.3V - 5V	3.3V - 5V	$17.95

The microcontroller subsystem of the jellybot is powered by the Freenove ESP32-WROOM and its compatible breakout board. These microcontrollers manage servo actuation through a Bluetooth-connected controller, enabling precise and responsive control. With input voltages ranging from 7-12V and output options of 3.3V and 5V, the system ensures stable power delivery to the servos while maintaining efficient communication and operation. This setup integrates seamlessly with the actuator subsystem, allowing dynamic and coordinated robot movements.

2.2. Software

JellyBOT utilizes two different software platforms: Raspi-code and Arduino. The Raspi-code software is responsible for controlling the camera, enabling real-time video feed for navigation and trash identification. The Arduino code, on the other hand, manages the operation of the lights and the robotic arm (claw) used for trash collection. This division of tasks allows for efficient control and seamless integration of the robot’s functionalities.

The Raspi-code is used to allow communication between the raspberry pi and the Raspberry Pi camera module.

Figure 2.1 Pseudocode for camera integration

```
# Import necessary libraries

Import PiCameraModuleLibrary

Import StreamingServerLibrary

Import TimeModule

Import OSModule

 # Function to set up the camera

Function setup_camera():

    Initialize Camera as camera_object

    Set camera_object.resolution = (Width, Height)

    Set camera_object.framerate = DesiredFramerate

    camera_object.brightness = DesiredBrightness

    camera_object.contrast = DesiredContrast

    Return camera_object

# Function to start live streaming

Function start_live_stream(camera_object, stream_port):

    Initialize StreamingServer as server_object with stream_port

    Create OutputHandler as handler_object using server_object

    Start camera_object.preview

    camera_object.start_recording(handler_object, format='h264')

    Print("Live stream started on port", stream_port)

# Main script logic

If __name__ == '__main__':

    Define Width = 1280
```

```
Define Height = 720

Define DesiredFramerate = 30

Define DesiredBrightness = 50

Define DesiredContrast = 50

Define StreamPort = 8000

camera = setup_camera()

Try:

    start_live_stream(camera, StreamPort)

    While True:

        Wait(TimeInterval)

    EndTry
```

The above pseudocode simplifies to the code to set up and fully integrate the camera into the robot. Initially it is connected to the raspberry pi and using Raspi-code, it is configured and set up to record video. Using python, the Raspberry Pi is programmed to allow for a simple stream that can be watched by all devices connected to the network. It can be stopped at any time but that is the most complex addition to the code.

The Arduino code is used to run all other subsystems as it is the most efficient code to use to control the robot as a larger system. The Raspberry Pi would need considerably better power and much more complex code to be able to control the rest of the robot and ultimately would create problems including a lack of visuals

2.2.1 Communication

The Arduino Nano microcontroller plays a key role in controlling the robotic trash collection arm of JellyBOT. It maps the inputs from a Nintendo Switch controller to the precise movements of the servo motors that operate the arm. The controller's joysticks and buttons are programmed to correspond to specific motions of the robotic arm, such as opening and closing the claw, extending or retracting, and rotating to grasp or release trash. When the user operates the Switch controller, the Arduino interprets the signals using predefined code and converts them into electrical pulses sent to the servo motors. These pulses, known as Pulse Width Modulation (PWM) signals, control the position and speed of the servos with precision. The Arduino’s compact design makes it ideal for the limited space on JellyBOT, while its versatility ensures smooth communication with the controller. By integrating the Arduino with the Nintendo Switch controller, the robotic arm achieves a high degree of responsiveness and accuracy, allowing the user to remotely manipulate the arm to efficiently collect underwater debris. This seamless interaction is critical for the robot’s effectiveness in varying aquatic environments.

3. EVALUATION

To ensure the JellyBot meets its purpose of cleaning underwater freshwater debris, extensive testing was conducted on each subsystem to verify compliance with the design constraints outlined in Table 1.1. These constraints encompass technical requirements, practical limitations, and adherence to engineering standards, all of which collectively ensure the functionality, safety, and efficiency of the JellyBot.

Table 3.1 - Technical Design Constraints

Name	Description
Buoyancy and Stability	The JellyBot achieves neutral buoyancy (tolerance ±10 grams) with materials and design that allow for stable, controlled movement underwater.
Waterproofing and Electrical Insulation	Waterproofing withstands depths of up to 3 meters (10 feet) for at least 30 minutes without any leakage.
Propulsion and Maneuverability	Thrusters provides a minimum thrust-to-weight ratio of 1:1, allowing JellyBot to overcome its weight in water and maneuver effectively.
Power Supply and Battery Life	All battery capacities allow for a minimum operational duration of 1 hour on a single charge. This will depend on individual power consumption.
Debris Collection Mechanism	The collection mechanism has a capacity to hold up a 16-ounce bottle without affecting the JellyBot buoyancy excessively

Each subsystem was rigorously tested to confirm that the JellyBot could operate effectively in underwater environments, maintain neutral buoyancy, withstand water pressure, and provide reliable maneuverability and control. The testing processes were designed to demonstrate how the robot meets its environmental goal of reducing pollutants in freshwater systems, thereby preventing their migration into saltwater ecosystems. By adhering to these constraints, the JellyBot fulfills its role as a robust solution for minimizing aquatic debris while prioritizing environmental and operational standards.

3.1. Test Certification – Propulsion

The testing of the propulsion subsystem began with verifying the operational voltage and electrical integrity of the chosen motors, the Submersible Boat Bilge Water Pump 12V 1100GPH Non-Automatic Marine Electric Bilge Pump (390 W), powered by a 12V rechargeable Li-Ion battery. Using a multimeter, each motor was tested individually to confirm proper voltage delivery and stable current flow under no-load conditions.

Electrical connections were first set up on a breadboard, where voltage drops were measured, and the circuits were checked for consistency. Any signs of irregular current flow were investigated to identify potential loose or faulty connections. With consistent readings across all circuits, the reliability of the electrical setup was confirmed, ensuring a solid foundation for further tests.

Once the motors were confirmed to operate at the correct voltage, the next step involved load testing to evaluate their performance over time. Each motor was connected to the propulsion system and tested under simulated operational loads while powered by the 12V rechargeable battery. The tests measured how long the motors could sustain operation without overheating or experiencing a noticeable drop in performance.

Table 3.2 - Marine Electric Bilge Pump Motors Load Test

Weight (lbs)	Thrust (lbs)	Thrust (N)	Thrust-to-Weight Ratio	Operating Voltage (V)	Voltage Change (V)
40	50	222.41	1.25	11.8	0.00
45	56	249.90	1.24	12.0	0.20
50	63	280.45	1.26	12.2	0.20
55	70	311.97	1.27	12.5	0.30
60	77	342.47	1.28	12.7	0.20

During this phase, the motors were also coupled with the remote-control subsystem, which included commands sent through the Flysky FS-i6 Model 2 remote control. This integration ensured that the motors responded seamlessly to directional inputs such as forward, backward, and lateral movements, as well as depth adjustments. Observations focused on the consistency and responsiveness of the commands, verifying that the remote-control system could provide real-time inputs without delays or interruptions.

The JellyBot was submerged in a water tank to evaluate its thrust-to-weight ratio, a critical factor for ensuring effective underwater performance. Testing across a weight range of 40-60 lbs demonstrated that the combined thrust consistently exceeded the required 1:1 ratio. This confirmed the ROV's capability to generate enough power for stable movement and precise control in underwater environments.

Table 3.3 - Thrust-to-weight Ratio Calculation

Weight (lbs)	Thrust (N)	Thrust-to-Weight Ratio
40	222.41	1.25
45	249.90	1.24
50	280.45	1.26
55	311.97	1.27

60	342.47	1.28

The following equation illustrates the calculation used to determine this ratio, highlighting the JellyBot's ability to meet performance expectations (1).

$$\text{Thrust-to-weight Ratio: Thrust} = \text{Weight} \times 4.45$$

During dynamic testing, the ROV executed ascending, descending, and rotating movements to assess its maneuverability and stability. Visible water streams demonstrated motor effectiveness, while the JellyBot maintained control without drifting or imbalance. These tests verified the ROV's ability to handle directional changes with precision.

Table 3.3 - Dynamic Testing: Stability and Motor Power

Movements	**Observed Drift (in)**	**Allowed Drift (in)**	**Stability Score (0-10)**	**Actual Thrust (N)**	**Max Thrust Capacity (N)**	**Motor Power (%)**
Ascending	0.20	6.00	9.97	222.41	300	74.14
Descending	0.08	6.00	9.99	249.90	300	83.30
Rotating	0.12	6.0	9.98	280.45	300	93.48

Throughout the tests, data on power consumption and thrust efficiency were logged. This analysis confirmed the propulsion system's reliability and suitability for underwater operations.

The following equations quantify the JellyBot's performance by calculating the stability score (2), which measures its ability to maintain balance and resist drifting during movements.

$$\text{Stability Score} = 10 \cdot \left(1 - \frac{Observed\ Drift}{Allowed\ Drift}\right) \tag{2}$$

The motor power measurements are represented by equation (3), highlighting the efficiency of the JellyBot's propulsion system in generating thrust relative to its maximum capacity.

$$\text{Motor Power} = \left(\frac{Actual\ Thrust}{Max\ Thrust\ Capacity}\right) \cdot 100 \tag{3}$$

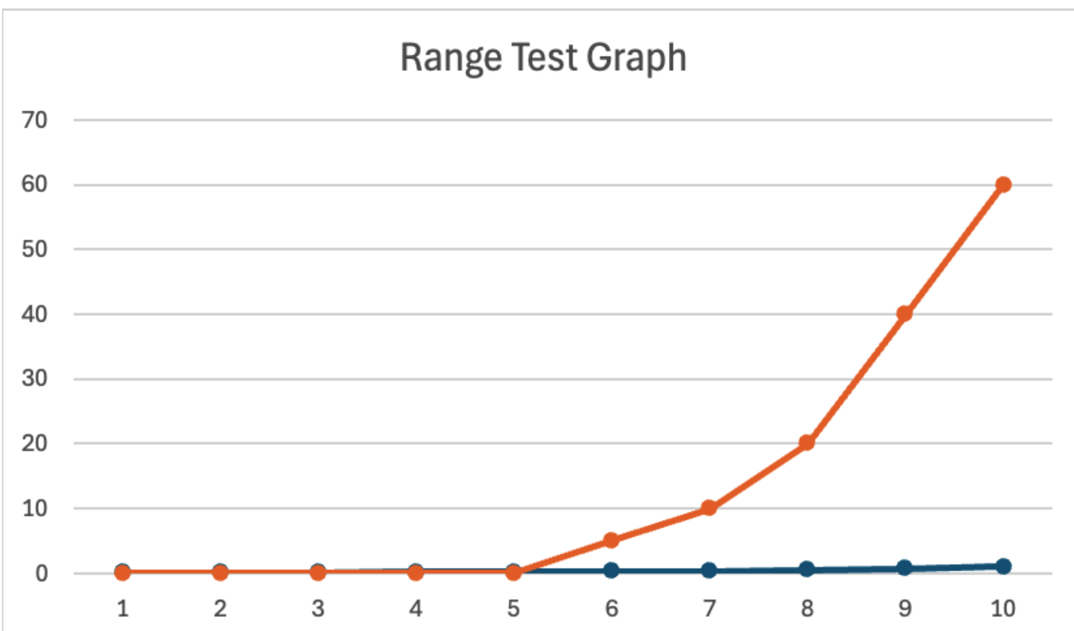
In addition to these tests, data on power consumption, motor performance, and control accuracy were collected and analyzed. Observations included ensuring that the propulsion system could maintain stable power under load, that motor responses to input commands were consistent, and that the overall system met the operational requirements outlined in the design constraints. This methodical testing confirmed that the propulsion subsystem is reliable, responsive, and well-suited for the JellyBot's underwater operations, ensuring it can effectively maneuver during debris retrieval tasks.

3.2. Test Certification – Remote Control

Testing for the remote-control subsystem evaluated the FS-i6 controller and the Nintendo Switch Pro controller with an ESP32 microcontroller and an Arduino nano. Both were tested for reliable communication, precise control, and sufficient range in dry and underwater conditions. The FS-i6 controller was calibrated to map joysticks and buttons to the JellyBot's functions, with channel configurations tested for directional control, depth adjustments, and stability.

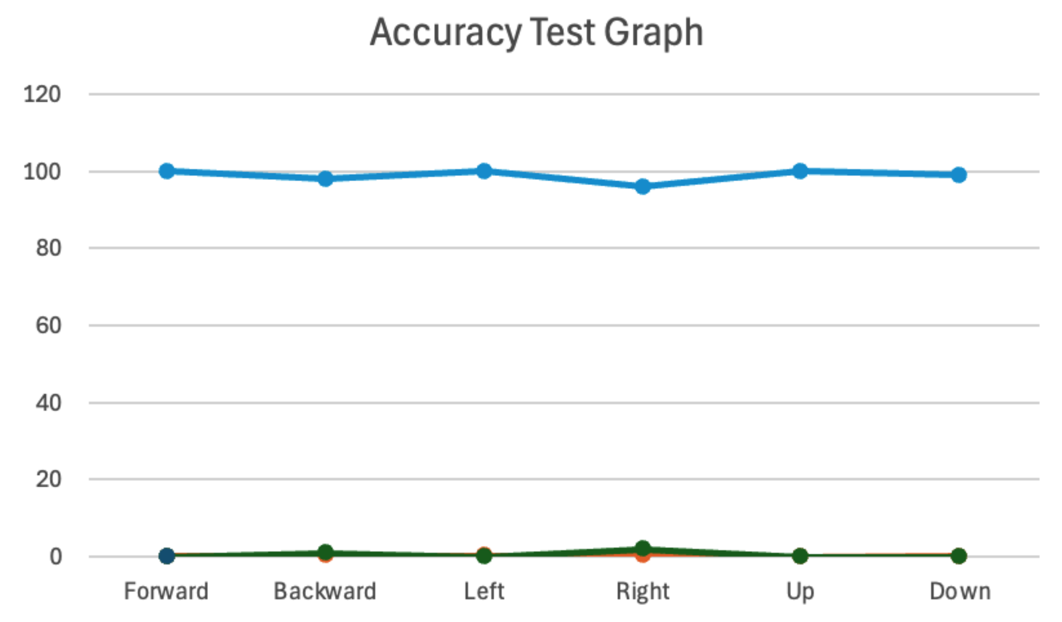
Range tests measured the FS-i6 controller's effective communication distance in an open environment by increasing the gap to the JellyBot and monitoring response. Latency and signal loss were recorded to determine the maximum reliable operating range under ideal conditions.

Figure 3.1 - Range Test Graph



The accuracy of the control commands was evaluated by sending directional inputs and monitoring JellyBot's response to each input. Delays were measured, and any unexpected movements were recorded. The system was observed for consistency in responding to repeated commands to ensure reliable performance.

Figure 3.2 - Accuracy Test Graph



Overall, the FS-i6 was more suited to the JellyBot's operational requirements in terms of range and underwater stability, though the Nintendo Switch controller offered a user-friendly interface for controlled environments where proximity was manageable. Future improvements could explore enhanced Bluetooth communication modules or underwater-rated signal boosters to extend the Nintendo controller's viability.

3.3. Test Certification – Servo Actuator

Using the Instructables guide as a foundation, the robotic arm was designed in Fusion 360 CAD software. Components such as the base, arm segments, and servo brackets were modeled for compatibility with servo dimensions and weight-bearing requirements. Custom mounting holes and wiring slots were incorporated, while joints were designed to allow a full range of motion without exceeding servo limits.

The parts were 3D-printed using durable materials like PLA or ABS to ensure strength for load-bearing components. After quality checks to verify adherence to design specifications, the parts were assembled with servos, wires, and support structures. Lubrication was applied to the joints to reduce friction and ensure smooth operation.

The Nintendo Switch Pro controller was paired with the ESP32 microcontroller, programmed to recognize and send signals to each servo motor in the robotic arm. Each button on the controller was mapped to specific movements of the arm, such as rotating the base, bending joints, or operating the claw.

Calibration ensured that button presses resulted in precise servo movements. Joystick sensitivity was adjusted for smoother transitions between speeds and positions, allowing for fine control over each joint's range of motion.

Each servo was tested individually to confirm its range of motion and smoothness, noting resistance or misalignment. Commands from the Nintendo Switch Pro controller were used to test response time and precision, with recorded delays and alignment against predefined positions.

Table 3.4 - Response Time and Precision Testing

Servo	Response Time (s)	Precision Score (0-10)	Adjustments Needed
Servo 1	0.15	9.8	None
Servo 2	0.18	9.7	Realigned
Servo 3	0.20	9.6	None
Servo 4	0.17	9.8	Tightened
Servo 5	0.16	9.9	None
Servo 6	0.19	9.7	Realigned

The servos demonstrated reliable performance, with response times between 0.15 and 0.20 seconds and precision scores ranging from 9.6 to 9.9. Minor adjustments, such as realignment and tightening, addressed misalignment issues, ensuring smooth and accurate movements for optimal robot arm operation.

In conclusion, the controller provided reliable, real-time input, with minimal lag observed for each servo command, especially in unloaded conditions. The arm successfully handled designated weights, though adjustments were noted for specific joints where the servos approached torque limits. Prolonged testing highlighted areas in the joints that might require reinforcement, as well as the need for periodic recalibration to maintain precision. The Nintendo Switch controller provided a user-friendly interface, although the effective range and Bluetooth stability were limited, especially if the ESP32 was submerged or positioned at a greater distance from the controller. Overall, the servo actuator subsystem, in combination with the Nintendo Switch Pro controller, demonstrated reliable functionality with manageable adjustments. Future improvements might explore increasing servo strength for heavier tasks and enhancing controller connectivity for broader operational flexibility.

3.4. Test Certification – Cameras

A raspberry pi camera module was initially only meant to be used for the showcase day. However, after researching the module as well as the raspberry pi, the decision was made to integrate the raspberry pi camera module into the overall robot. Basing most of the modification and initial installation instructions on resources such as YouTube, instructables, and the raspberry pi usage forum, the camera module was fully integrated into the robot with minimal issue.

Figure 3.1 - Camera Integration

The camera's lens was extended to allow for sight farther away. It was also placed into a waterproof case with only the ribbon cable exposed to the water. The raspberry pi itself was placed into a waterproof box and the battery pack powering it was also placed into a waterproof box. All connections were sealed to deter any water damage to the camera system.

Step 1: Pre-integration Testing

The camera was initially set to have its own power source and only recorded video. For the initial demonstration the camera worked perfectly and had minimal inconsistencies other than taking a bit of time to process movement.

Step 2: Post Modification Testing

The camera's video feed was changed from a saved video to a stream that could be accessed via phone hotspot by multiple devices. If all devices connected to it are connected to the same Wi-Fi network, they will be able to watch the stream in real time.

Step 3: Waterproofed testing

Testing after the camera was in the case was simple. The camera, raspberry pi, and power bank were all held a faucet and then eventually completely submerged in water. The robot still worked in both settings with the only detrimental effect being a slightly spotty image while it was under the faucet. While it was fully submerged, the robot worked sufficiently.

Step 4: Post Integration Waterproofed Testing

Post integration waterproof testing was also simple. The test consisted of placing the robot into a small body of freshwater and running the stream as it went underwater. This would allow for a full visual upon entry into the water as well as allowing for the visualization of any potential issues with the camera upon submersion.

Step 5: Post Testing Modifications

The main modifications made post integration and testing are waterproofing the entire robot and redoing any waterproofing that was in place that may have been damaged. The stream has shown compatibility with all electronics even when fully submerged. The battery has also been extended to the shore to lessen the likelihood of it being drowned after early testing of the waterproofing showed that there was a high risk of damage to the power supply due to faulty waterproofing.

3.5. Test Certification – Power

The wireless charging module was tested to evaluate its performance in real-world conditions. The transmitter and receiver were positioned within the ROV's operating distance, and voltage and current outputs were measured to ensure they matched the system's requirements. Observations focused on maintaining stable voltage output, as fluctuations could affect electronic stability.

Efficiency testing involved measuring the power input to the transmitter and output from the receiver over time. Efficiency was calculated, with results revealing significant power losses at increased distances. These measurements highlighted the importance of precise positioning to maximize energy transfer.

Table 3.5 - Efficiency, Distance, and Alignment Testing Table

Distance (in)	**Power Input (W)**	**Power Output (W)**	**Efficiency (%)**
0.00	10	9.5	95.00
1.97	10	9.0	90.00
3.94	10	8.5	85.00
5.91	10	7.0	70.00
7.87	10	5.0	50.00

The module was tested at various distances and orientations to determine optimal placement. Increasing distance reduced charging efficiency, with charging ceasing entirely beyond 20 cm. Positional tolerances were also observed, where misalignments caused notable power loss, emphasizing the need for precise receiver mounting in the ROV enclosure.

3.6 Testing Certifications – Battery

Each battery was tested for its capacity, discharge rate, and charging performance to ensure it could meet the ROV's power demands reliably. The procedures focused on verifying operational reliability and identifying any inefficiencies or inconsistencies in battery performance.

Testing the 12V rechargeable batteries covered capacity verification, voltage stability, and recharge performance. Capacity tests measured discharge times under controlled loads to confirm operational duration, noting discrepancies from rated capacity. Voltage stability was assessed with a multimeter underload to detect any significant drops. Recharge tests recorded charging times and monitored for excessive heat, identifying inefficiencies or internal resistance issues.

Table 3.6 - Voltage Stability, Capacity Verification, and Recharge Rate & Heat Dissipation

Test	**Procedure**	**Observation Focus**
Voltage Stability	Measured periodic voltage under load using a multimeter to ensure stability.	Noted sudden voltage drops indicating output issues.
Capacity Verification	Charged and discharged batteries under controlled loads to measure operational duration.	Compared actual vs. rated capacity for discrepancies.
Recharge Rate & Heat Dissipation	Recorded recharge times and monitored temperatures during the process.	Identified unusual heating patterns and inefficiencies.

The table outlines key procedures and observations for each test. Voltage stability tests checked for sudden drops, capacity verification compared actual vs. rated performance, and recharge rate tests monitored charging times and heat dissipation. These evaluations ensured the batteries' reliability for the ROV.

The 3000 mAh DC 12V Li-Ion battery serves as a smaller power source for the ROV's auxiliary systems, such as lights and sensors. Its compact size and energy capacity make it suitable for low-power tasks, ensuring efficient operation without overloading the ROV's main power supply. Testing was conducted to evaluate the battery's ability to meet these demands reliably.

The battery was tested for runtime by powering the ROV's subsystems until depletion, recording a minor discrepancy between expected and actual durations. Voltage regulation ensured a stable 12V output under varying loads, with occasional minor dips noted. Thermal performance was assessed by monitoring heating patterns, with temperatures remaining within a safe range during extended discharge.

Table 3.7 - Voltage Regulation, Thermal Performance, and Load Testing & Runtime

Test	**Measurement**	**Observations**

Load Testing & Runtime	Runtime: 120 minutes (expected 130 minutes)	Minor runtime discrepancy due to increased subsystem demand.
Voltage Regulation	Voltage: 12.0V ± 0.2V under varying loads	Stable voltage output with occasional minor dips.
Thermal Performance	Max Temperature: 45°C (no hot spots detected)	Temperature remained within safe range, no overheating.

The table above provides an overview of the battery's performance, including runtime, voltage stability, and thermal behavior. It highlights minor discrepancies in runtime, stable voltage output with occasional dips, and safe thermal performance without signs of overheating, demonstrating the battery's reliability for auxiliary power needs.

The 30000 mAh power bank was tested for capacity by fully discharging it through a controlled load that simulated the ROV's operational environment. The duration it sustained the load was recorded and used to calculate the effective capacity, which was then compared to the rated 30000 mAh. This test verified whether the power bank met expected endurance levels, ensuring it could support extended ROV missions.

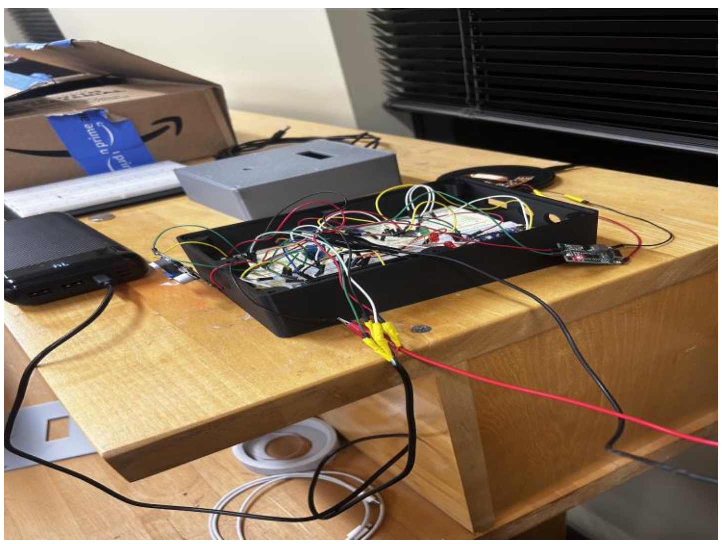


Figure 3.2 - Wireless Charging Connection to 30000 mAh Power Bank

To assess output voltage consistency, the power bank's USB output was monitored using a digital oscilloscope. This test identified any fluctuations that might disrupt connected electronics, particularly those requiring stable 5V or 12V outputs. Observations highlighted any voltage instability under load, which could compromise the performance of sensitive ROV components.

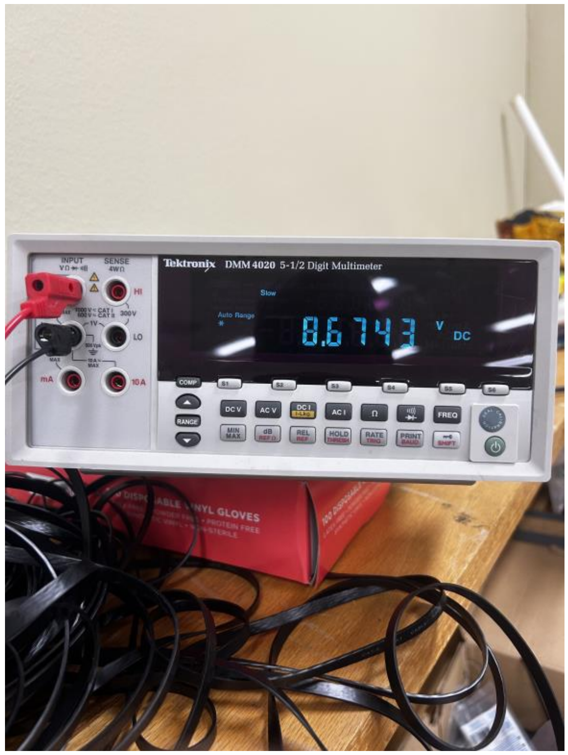


Figure 3.3 - Voltage Regulation for Wireless Charging Module

The power bank's ability to charge multiple devices simultaneously was evaluated by connecting it to systems like the wireless charging module and Li-Ion battery. Each output port was monitored for consistent power delivery without dropouts or significant voltage drops. Observations included verifying adequate power for all connected devices and checking for activation of over-current protection features to ensure reliable operation.

The wireless charging module demonstrated reliable performance within specific distances and alignments, with efficiency slightly reduced by minor misalignments, emphasizing the need for precise mounting. While thermal stability was generally acceptable, mild heating during extended use suggested the importance of monitoring during long missions. The 12V rechargeable batteries provided stable power under load, though slight capacity differences indicated potential future replacements for aging cells. Charging times were within expectations, but cooling periods were recommended after intensive discharges to maintain performance.

The 3000 mAh DC 12V Li-Ion battery reliably powered low-load subsystems with stable voltage and safe thermal performance but lacked capacity for high-drain tasks. Testing confirmed the power subsystem's reliability, ensuring the ROV's readiness for underwater missions.

4. **SUMMARY AND FUTURE WORK** [minimum length: ½ page]

Throughout the development of JellyBot, the team successfully accomplished several key objectives, resulting in a functional underwater robot designed for trash collection in freshwater environments. The PVC frame proved to be a durable and cost-effective solution, providing the necessary structural integrity for the robot's underwater operations. The propulsion system, powered by propellers, allowed the robot to maneuver effectively in aquatic environments, enabling efficient movement and depth control.

Additionally, the robotic arm, controlled via servos, successfully performed the trash collection function, showcasing the robot's ability to grasp and handle various objects underwater.

One of the most successful aspects of JellyBot was its movement system. The use of the Flysky FS-i6 remote control to manage the robot's three motors provided precise control over its movement and depth. This system allowed the robot to navigate through underwater environments with ease, offering smooth adjustments to direction and positioning. The motors responded well to commands, and the robot demonstrated the ability to travel up to 12 feet underwater, which was critical for the project's goals. The control system was flexible and allowed for detailed adjustments, making it an essential part of the robot's functionality. The combination of efficient propulsion and responsive motor control allowed JellyBot to move efficiently in a range of underwater conditions, making movement one of the key successes of the project.

Future extensions for JellyBot would focus on improving its autonomy and enhancing its environmental awareness. Adding depth sensors would allow the robot to maintain precise control over its movement and depth, preventing potential collisions or getting stuck. These sensors would enable JellyBot to automatically adjust its depth based on environmental conditions, such as varying water levels. Moreover, integrating proximity sensors or sonar technology would improve the robot's ability to navigate its surroundings, detect obstacles, and avoid damage, especially in cluttered or complex underwater environments.

Additionally, improvements to the robotic arm would be beneficial. Upgrading to more advanced servo motors or actuators could increase the precision and strength of the arm, enabling it to handle a wider variety of trash more effectively. The arm could also be enhanced with the ability to adapt to different types of debris, potentially incorporating machine learning algorithms to identify objects and prioritize collection based on size, shape, or material.

One of the changes the team had to make during the development of JellyBot was to the collection system. Initially, the goal was to incorporate a trash collection system that would allow the robot to store waste in a compartment, enabling it to continue its journey without needing to return to the surface for disposal. This system would have been advantageous for the user, as it would maximize the robot's time in the water by allowing it to collect trash while still seeking out more debris. However, the team encountered several challenges in implementing this idea, including limitations in resources, time, and budget. As a result, the collection system was not fully integrated into the final design. Despite this, the team still considers adding a collection system to JellyBot in the future as a key goal for later development.

Further, while the Raspberry Pi Camera module provided adequate visual input, additional sensors could be incorporated to improve JellyBot's decision-making capabilities. For example, temperature sensors could provide real-time data on the water's temperature, and pressure sensors could help the robot better understand the underwater environment, adapting its behavior based on changes in pressure and other environmental factors.

In conclusion, while JellyBot succeeded in achieving key goals such as mobility and trash collection, there are several areas where improvements could enhance the robot's functionality. By adding environmental sensors, enhancing autonomy, and improving the robotic arm, future iterations of JellyBot could become more efficient, autonomous, and capable of handling a wider variety of tasks in dynamic and challenging underwater environments. These enhancements would allow JellyBot to better fulfill its mission of reducing underwater trash accumulation in freshwater environments.

5. ACKNOWLEDGEMENTS

We wish to acknowledge Timothy Sellers, who acted as both our professor for Senior Design 2 as well as our advisor. We would also like to acknowledge Dr. Beck for her support in the conception and completion of the JellyBot. Dr. Beck's interactions have helped us simplify our overall design as well as add features to some of our subsystems, much of which aided in our development of the JellyBot.

6. REFERENCES

[References must be formatted precisely according to current IEEE guidelines located at http://ieeeauthorcenter.ieee.org/wp-content/uploads/IEEE_Style_Manual.pdf#page=36.]

[1] S. Larky, et al, “You’ve come a long way, USB: history and future of USB port technology,” Electronic News, July 2001. [Online]. Available: http://findarticles.com/p/articles/mi_m0EKF/is_30_47/ai_76780475. [Accessed: Sept 7, 2007].

[2] J. M. D. Pereira, “The history and technology of oscilloscopes: an overview of its primary character istics and working principles,” IEEE Instrumentation and Measurement Magazine, pp. 27-35, December 2006. [Online]. Available: http://ieeexplore.ieee.org. [Accessed: Sept. 4, 2007].

[3] Bagley College of Engineering, “Computer Requirements Overview,” Bagley College of Engineering, 2007. [Online]. Available: http://www.engr.msstate.edu/prospective_students/computer_initiative. [Accessed: Sept. 5, 2007].

[4] M. T. Gibbons, “Engineering by the Numbers,” American Society for Engineering Education, 2007. [Online]. Available: http://www.asee.org/publications/profiles/upload/2006ProfileEng.pdf. [Accessed: Sept. 4, 2007].

[5] Tektronix, “Tektronix Company Overview: Markets, Strategy, and Results,” Tektronix Inc., May 2007. [Online]. Available: http://library.corporateir.net/library/81/815/81500/items/244554/Company_Presentation_2007_v3.pdf. [Accessed: Sept. 5, 2007].

[6] R. A. Witte, Electronic Test Instruments: Analog and digital measurements, 2nd ed. Upper Saddle River: Prentice Hall, 2002, pp. 79-134.

[7] Cypress Semiconductor Corporation, Datasheet, CY8C21123, CY8C21223, and CY8C21323, PSOC Mixed-Signal Array Final Data Sheet.

[8] Microchip Technology Inc., Datasheet, PIC18F2455/2550/4455/4550, 28/40/44-Pin, High Performance, Enhanced Flash, USB Microcontrollers with nanoWatt Technology.

[9] Cypress Semiconductor Corporation, Datasheet, CY7C024-15AXC, 4K x 16/18 and 8K x 16/18 Dual- Port Static RAM with SEM, INT, BUSY.

[10] Xilinx Inc., Appl. Note 463, p. 1.

[11] “Design Guide for Xilinx FPGA Power Management Systems,” October 2004, dkc3.digikey.com/PDF/Marketing/FPGA_Xilinx.pdf.

[12] National Semiconductor Corporation, Datasheet ADC08200, 8-Bit, 20 MSPS to 200 MSPS, 1.05 mW/MSPS A/D Converter with Internal Sample-and-Hold.

[13] National Semiconductor Corporation, Datasheet, ADC08060, 8-Bit, 20 MSPS to 60 MSPS, 1.3 mW/MSPS A/D Converter.