

Modular Submersible Vehicle with Autonomous Navigation

Rodrigo Aguilar, Spencer Chen, Philip Huang, Stanley Kim | Acknowledgements: Dr. Romel Gomez, Jay Rana, Nolan Ballew

Problem Definition

Objective

1. Identify an object underwater
2. Approach the object
3. Determine the object's location
4. Report the object's coordinates

Constraints

- Must search inside a 3' x 3' x 7' aquarium
- Low power thrusters: <1 W power draw
- Battery specifications: 7.4 V, 2000 mAh capacity, 6 A max draw
- Cost: <\$1,000

Background & Motivation

Oceans cover approximately **70%** of the Earth's surface yet our understanding of underwater ecosystems is limited due to lack of accessibility. Current methods such as seafloor mapping provides a sense of what may lie beneath and guides decisions about where to explore, but they can't identify species on the seafloor.

An autonomous submersible vehicle like the one we built could be used for many applications, such as identifying species, observing animal behaviors, mapping out habitats, monitoring ecosystem health such as coral reefs and kelp forests, and exploring difficult-to-reach areas.

Vehicle Characteristics

Current Limitations of Existing Vehicles

- Traditional ROVs are bulky and slow due to heavy instrumentation
 - Poor performance in tight, complex environments
 - Limited ability to track moving organisms
- Many ROVs rely on cables for power + communication
 - Limited range and maneuverability
 - Risk of cable entanglement

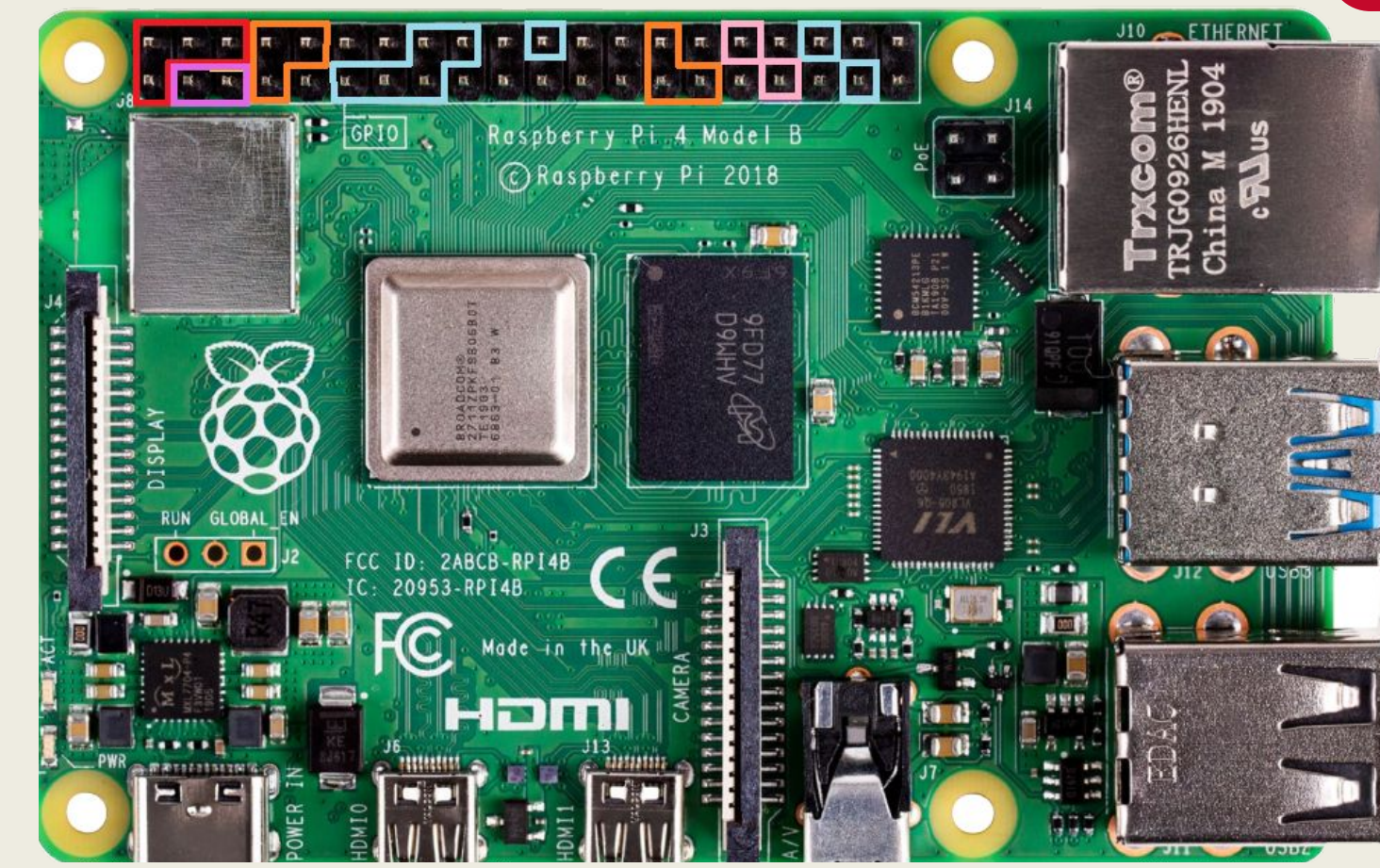
Our Advancements

- Compact, Self-Contained, and Maneuverable Design
 - Can move forward/back + rotate precisely
 - Small frame and lack of wired connection allows easier navigation of tight spaces (reefs, caves, wrecks)
 - Wireless connection via SSH allows for data transfer
- Computer Vision
 - Provides real-time detection and tracking of moving objects such as unknown species
 - Removes the need of a human operator for vehicle control

Development Process

Date	Progress
3/5	Preliminary Design Finalized
4/2	Achieved Manual Depth Control
4/9	Implemented Computer Vision & Object Detection
4/14	Achieved Automatic Depth Control
4/23	Achieved Automatic Centering
4/28	Completed Full Navigation Process

Raspberry Pi 4 Model B



- Wiring Legend**
- 3.3V, 5V, GND
 - Sensor UART
 - Motor OUT
 - IMU I2C
 - Motor PWM
- Raspberry Pi Camera 3**
- Provides live video capture for object detection.

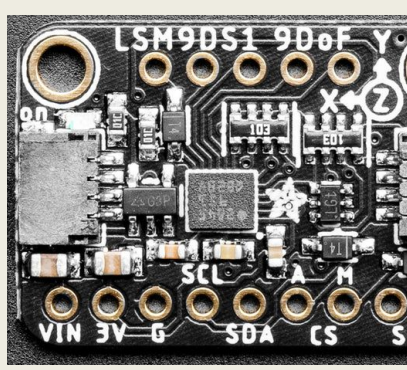
Components



- RRS20 IP68 Switch**
- Allows vehicle to be powered on or off without disassembly.



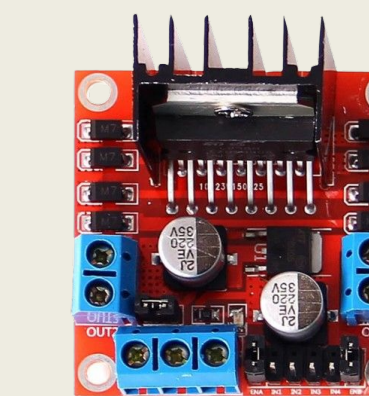
- SEN0598 Distance Sensor (x3)**
- Reports underwater distance up to 3 meters.



- Adafruit LSM9DS1 IMU**
- Reports the current orientation (yaw) of the vehicle.

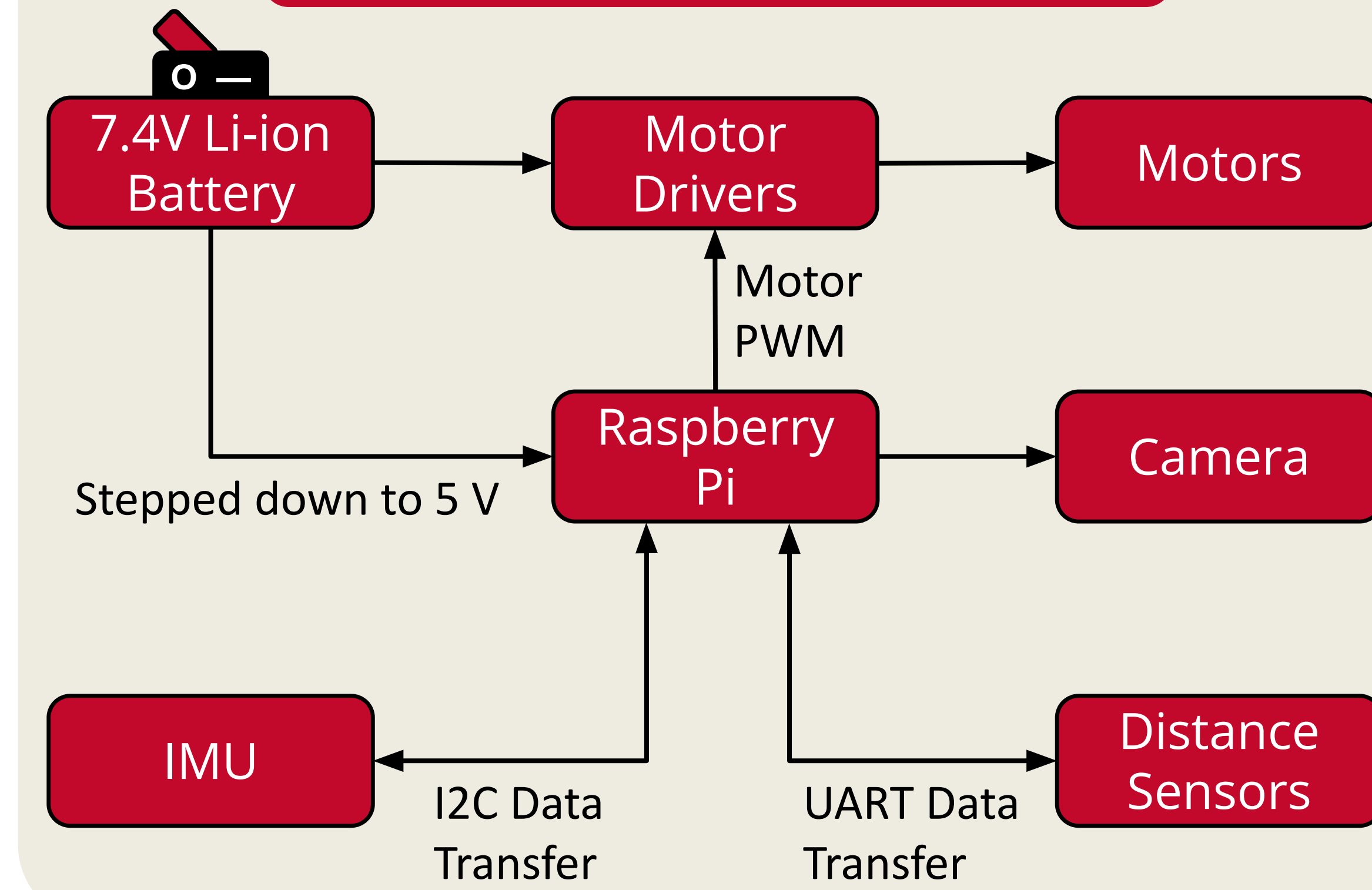


- BM70 Underwater Motor (x4)**
- Provides thrust.

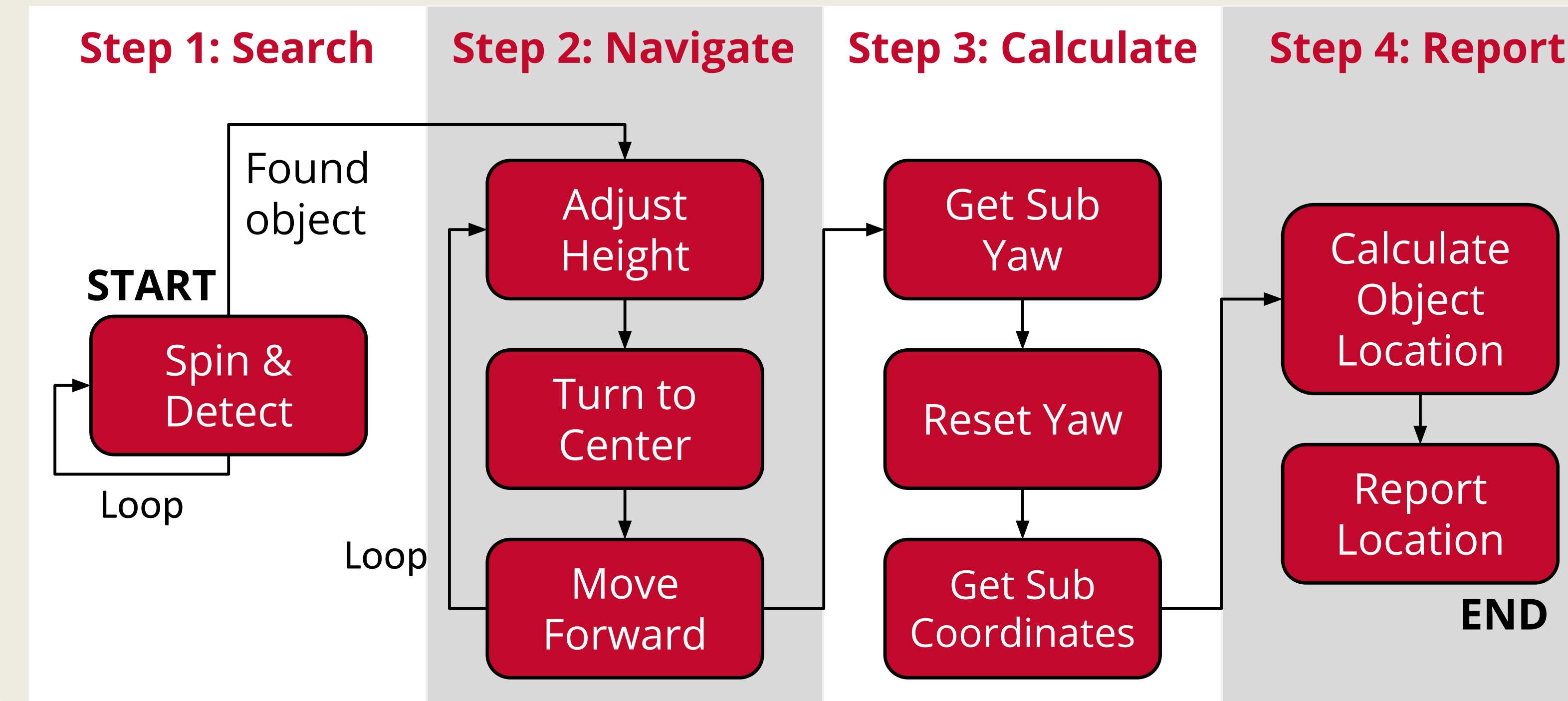


- L298N Motor Driver (x2)**
- Allows for speed and direction control of two DC motors at once.

System Diagram



State Diagram

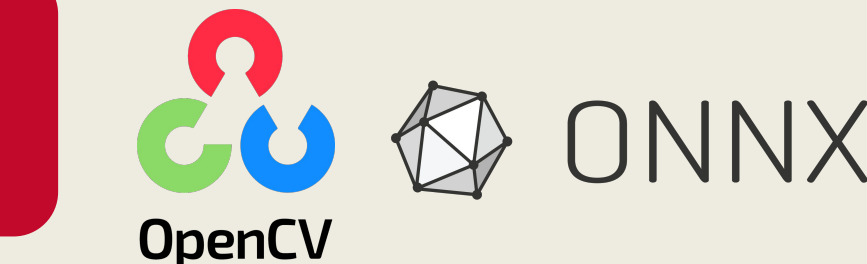
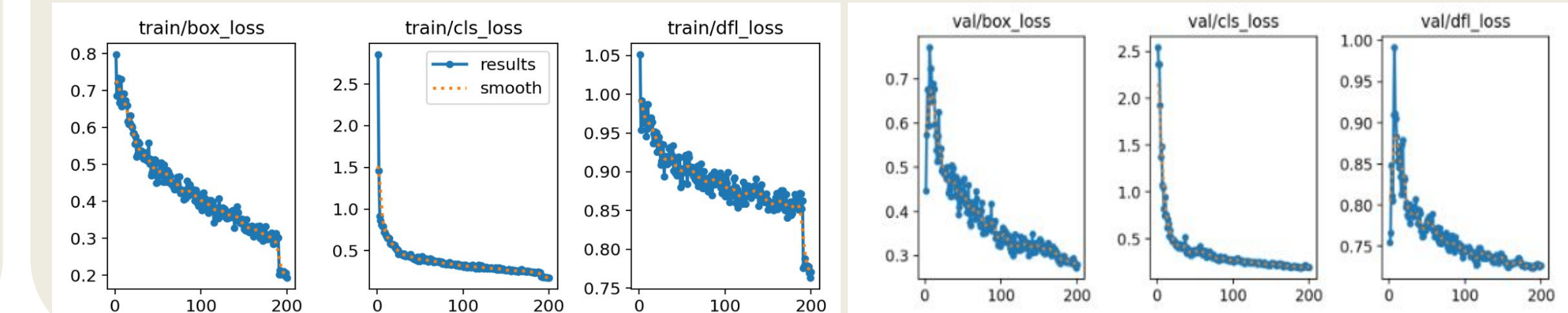


Computer Vision

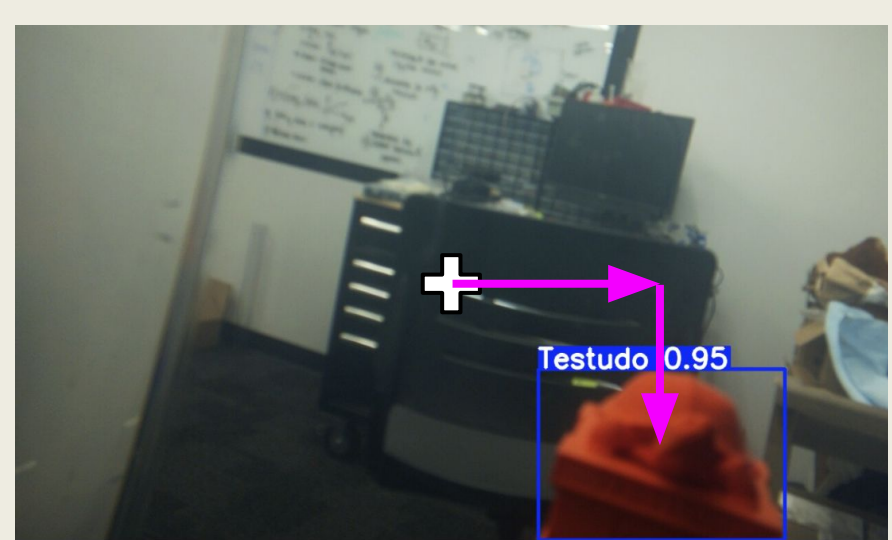


Training through Ultralytics YOLO11 Nano

- 1) **Taking 300 pictures** underwater with Raspberry Pi Camera Module 3
- 2) **Label Studio** to import and manually annotate all 300 pictures with desired target class: "Testudo"
- 3) **Separating into training and validation data**, (224 training images + 25 validation images)
- 4) **Training the model with appropriate parameters**: YOLOv11 Nano model, 200 epochs, image size of 320 x 320 pixels, patience of 50, and augmentation

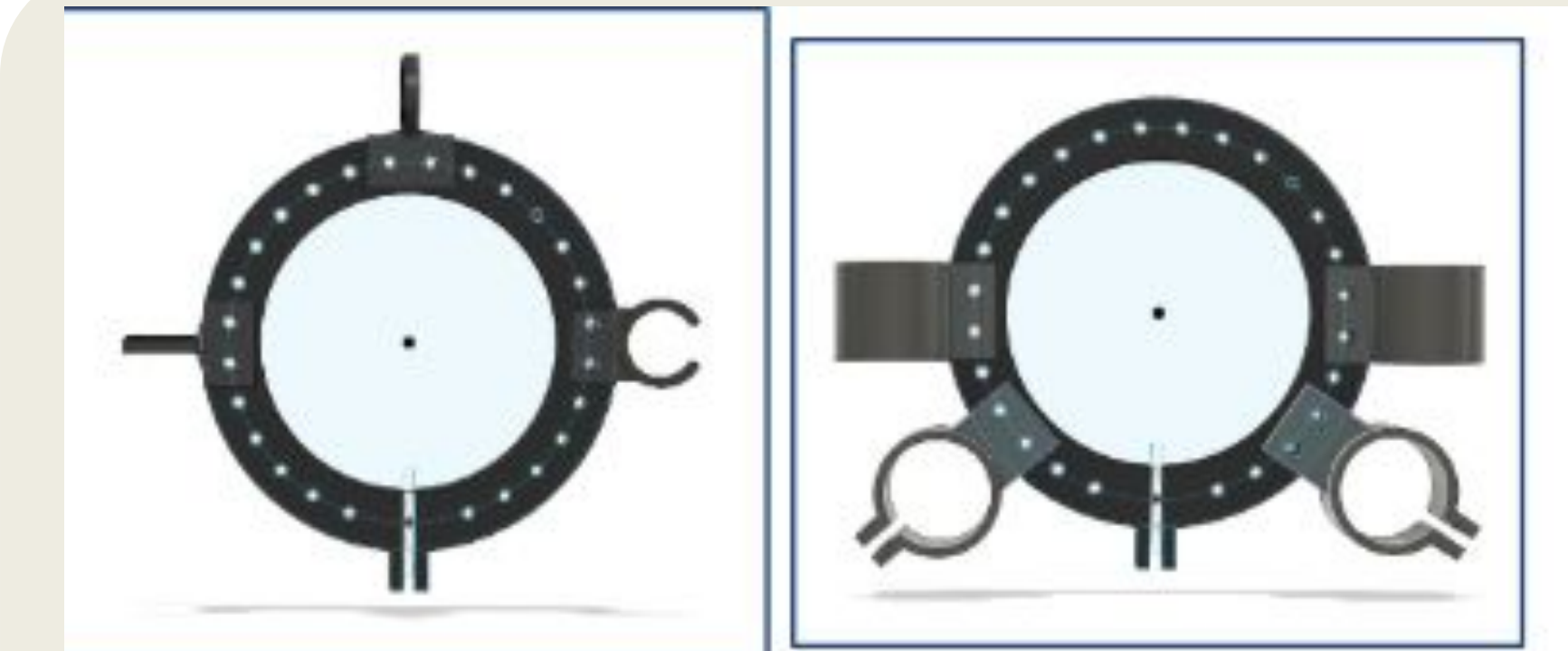


Submarine Camera View



Sub will move **down** and turn **right**

Final Design



Modular Mounting Rings, Sensor Mounts, Motor Mounts

- 28 holes standardized to 8-32 screws
- Easy adjustment along the Tube body → Center of Mass adjustments
- Allows for various mount configurations

Utilization of CAD → Fully 3D Printed Mounts

→ Cost and Time efficiency of mount replacement if damages

3D Printed Internal Slide + Detachable Weights

- 0.534 in x 0.736 in Middle Column, fits double stacked
- 0.21 in x 0.71 in adhesive 1 oz weights
- Accounting for adjustable balancing of buoyancy in depth the roll and tilt orientations

