Navigating Across Domains: AI-Enabled Cooperative Autonomy for USV–UAV Systems

Xavier Vicent
Department of Ocean & Mechanical Engineering
Florida Atlantic University
Dania Beach, FL, USA
xvicentnavar2024@fau.edu

Daniel Morris
Department of Ocean & Mechanical Engineering
Florida Atlantic University
Dania Beach, FL, USA
danielmorris2022@fau.edu

Manhar R. Dhanak
Department of Ocean & Mechanical Engineering
Florida Atlantic University
Dania Beach, FL, USA
dhanak@fau.edu

Samantha Garcia
Center for Connected Autonomy & Artificial Intelligence
College of Electrical Engineering & Computer Science
Florida Atlantic University
Boca Raton Beach, FL, USA
scruz10@fau.edu

Clarck Dorcent
Center for Connected Autonomy & Artificial Intelligence
College of Electrical Engineering & Computer Science
Florida Atlantic University
Boca Raton Beach, FL, USA
cdorcent2021@fau.edu

Adriel Marie Abad
Center for Connected Autonomy & Artificial Intelligence
College of Electrical Engineering & Computer Science
Florida Atlantic University
Boca Raton Beach, FL, USA
aabad2022@fau.edu

Bob Varghese
Center for Connected Autonomy & Artificial Intelligence
College of Electrical Engineering & Computer Science
Florida Atlantic University
Boca Raton Beach, FL, USA
bvarghese2026@fau.edu

Georgios Sklivanitis
Center for Connected Autonomy & Artificial Intelligence
College of Electrical Engineering & Computer Science
Florida Atlantic University
Boca Raton Beach, FL, USA
gsklivanitis@fau.edu

Dimitris A. Pados
Center for Connected Autonomy & Artificial Intelligence
College of Electrical Engineering & Computer Science
Florida Atlantic University
Boca Raton Beach, FL, USA
dpados@fau.edu

***Abstract*—This paper presents the design and competition strategy of Team SeaOwls for the 2026 Maritime RobotX Challenge, centered on an integrated Uncrewed Surface Vehicle (USV)–Uncrewed Aerial Vehicle (UAV) System of Systems (SoS). The architecture combines AI-based perception, cooperative autonomy, communication, planning, and control to support multi-domain mission execution. The USV integrates camera–LiDAR sensor fusion for object detection and localization, an Artificial Potential Field (APF) planner for collision-aware navigation, and a Deep Reinforcement Learning (DRL) controller trained using Proximal Policy Optimization (PPO). The UAV contributes aerial perception, communication, and connected autonomy to extend situational awareness and support cooperative tasks. A ROS 2-based software framework enables information sharing between vehicles and integration with the RobotX RoboCommand system. The overall design emphasizes reliability, modularity, and cross-domain cooperation, supported by simulation, data-driven validation, and field testing.**

***Index Terms*—Maritime robotics, autonomous surface vehicle, uncrewed aerial vehicle, System of Systems, artificial intelligence, sensor fusion, cooperative autonomy.**

I. Introduction

Integrating Artificial Intelligence (AI) into the perception, modeling, control, and coordination of autonomous vehicles is a primary objective of the FAU RobotX team in developing a reliable System of Systems (SoS) for the 2026 Maritime RobotX Challenge. This year's competition places greater emphasis on cooperative, multi-domain autonomy, requiring multiple uncrewed systems to share information and work together toward a common mission rather than operating as independent platforms.

For RobotX 2026, the FAU team is formed through the collaboration of two student organizations: the Marine Robotics Club (MRC) and the Center for Connected Autonomy and AI (CA-AI) Drone Club. This partnership combines experience in autonomous surface vehicles, UAVs, computer vision, control, and AI to develop an integrated USV–UAV system. The team focuses on reliable perception, navigation, inter-vehicle communication, and coordinated decision-making so that both platforms can contribute to a common autonomous mission.

MRC brings experience from previous RoboNation competitions and a strong background in the maritime domain, while the CA-AI Drone Club contributes fresh innovation in connected autonomy, aerial systems, and AI. Together, the team continues to incorporate AI-based methods into object detection, localization, vehicle modeling, and autonomous control. By combining surface and aerial sensing within a shared System of Systems (SoS) architecture, the team aims to improve situational awareness, system robustness, and cross-domain cooperation throughout the 2026 competition.

II. Competition Strategy

For RobotX 2026, the team focuses on deploying a reliable USV–UAV *System of Systems* (SoS) capable of cooperative perception, planning, and autonomous decision-making. On the USV, camera and LiDAR data are fused for obstacle detection and localization, while information from the UAV provides additional aerial situational awareness. An Artificial Potential Field (APF) planner generates collision-aware trajectories and adapts the USV path as new information becomes available.



Fig. 1. SeaOwls Team at RoboBoat 2026.

The strategy prioritizes tasks that directly exploit USV–UAV cooperation, particularly *Safe Passage* and *Coordinated Logistics*, while maintaining a common autonomy architecture across the course.

The UAV uses onboard computer vision processing to detect and classify targets which it encodes before sending the detections to the control station. These encoded messages are decoded at the control station and converted to ROS topics to send back to the USV. The UAV also uses perception data to autonomously navigate to designated locations, deliver required resources, and return to its launch point.

A. Course Approach

The USV and UAV operate within a common ROS 2-based autonomy architecture. Camera–LiDAR perception, UAV-shared information, APF-based planning, and vehicle control are reused across tasks, while task-specific behaviors are implemented at the mission level. This modular approach reduces system complexity and allows validated autonomy components to be applied throughout the competition.

B. Task Selection in Scoring and Finals

During scoring rounds, tasks are selected based on execution reliability, system state, and communication readiness. *Safe Passage* and *Coordinated Logistics* are prioritized because they directly demonstrate cooperative autonomy between the USV and UAV.

In the finals, the strategy shifts toward continuous mission execution, with task sequencing considering vehicle state, UAV endurance, perception confidence, communications, and task proximity to reduce unnecessary transitions and improve mission reliability.

C. Situational Awareness and Communications

A graphical user interface (GUI) at the Operator Control Station (OCS) displays USV and UAV status, system health, and current mission tasks. The OCS also interfaces with RoboCommand to transmit vehicle heartbeats and task reports while receiving competition commands.

D. Task 1 – Safe Passage

Safe Passage is a primary demonstration of USV–UAV cooperation. The USV uses camera–LiDAR perception and APF planning to navigate the buoy field. At higher capability tiers, the UAV detects the safe route from above and communicates relevant buoy and hazard information to the USV. Real-time UAV updates can then be incorporated into the USV planner to adapt the route as conditions change.

The UAV conducts aerial search using a rapidly exploring random trees (RRT) and deep learning (DL) based algorithm to search and detect beacons in the field. A multi-tier system is used to first detect and identify the beacon, determine the beacon color, and then determine its blinking status. A TensorFlow Lite (TFLite) model is used to identify the beacon from the front camera stream, which is then cropped before running a second model that identifies the top of the beacon. Image processing is used on images of the beacon top to determine the lit area and color. With intrinsics from a calibration configuration (similar to OpenCV's ArUco pose-estimation workflow), the beacon's location respective to the drone is approximated using the known dimensions of the beacon, the altitude of the drone determined by the drone pose, and the bounding box of the beacon in the detected image. This is used alongside the UAV's GPS location to geolocate the beacon. Appendix H shows preliminary design and testing for the beacon detection model.

The beacon color, blinking status, and location are encoded and sent back to the UAV control station via MAVLink, which is then decoded and converted to a ROS topic to be sent to the USV control station.

E. Task 2 – Infrastructure Survey and Repair

Because this task primarily requires UUV–UAV cooperation, it is not a primary objective for the team's USV–UAV configuration.

F. Task 3 – Coordinated Logistics

For Coordinated Logistics, the USV detects the available docking bay, approaches and docks autonomously, and identifies the corresponding resource request. The UAV supports the task through inter-vehicle communications, receiving and relaying task information within the SoS. The UAV configuration is modified for this task to mount a robotic gripper for payload pickup and recovery based on information received from the USV. Based on currently validated systems, the team's strategy focuses on demonstrating reliable USV perception, docking, and cross-domain communication with a secondary focus on UAV payload pickup and recovery.

G. Task 4 – Dynamic Incident Response

Dynamic Incident Response tests the SoS ability to adapt to unexpected mission changes received through RoboCommand. The USV can interrupt its current objective, incorporate new keep-out zones or moving hazards into the APF planner, and resume mission execution after completing the requested response. This task demonstrates adaptive planning, communications, and autonomous re-tasking.

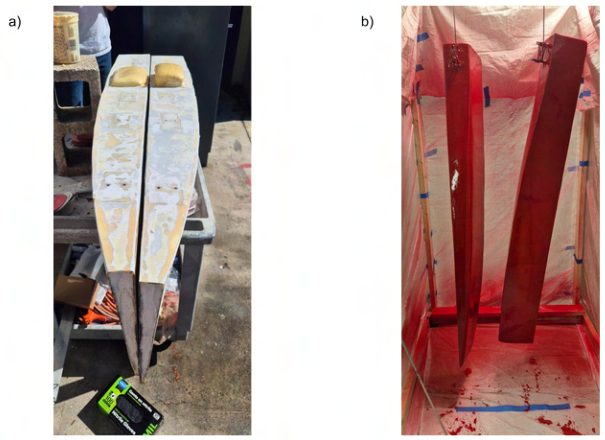


Fig. 2. Building the hull.



Fig. 3. USV on the water doing testing.

III. Design Strategy

The RobotX 2026 USV builds on the platform developed for RoboBoat 2026 [1], with modifications focused on reliability, increased payload capacity, and integration with the UAV as part of the team's System of Systems (SoS).

The UAV design strategy used parallel development of a custom aircraft and a commercial off-the-shelf (COTS) platform. The custom UAV was developed to meet specific payload, sensing, and communication requirements while the COTS UAV provided a proven and rapidly deployable alternative. Maintaining both systems helped to reduce development risk and provide redundancy if either aircraft failed.

A. USV Hull and Propulsion

The USV hull was redesigned after previous competition experience revealed insufficient freeboard, difficult-to-repair leaks, and structural limitations under increased payload. The new asymmetrical catamaran configuration increases displacement and transverse stability while providing improved access to propulsion and electronics components.

Propulsion was upgraded from Blue Robotics T200 to T500 thrusters to provide greater thrust authority under wind and environmental disturbances. Two stern-mounted thrusters provide differential thrust, reducing mechanical and control complexity while maintaining the maneuverability required for navigation and docking tasks.

B. USV: Guidance, Navigation, and Electronics

A new centralized electronics architecture replaces the larger system inherited from the previous WAM-V platform. Safety-critical power components are separated from computing and sensing hardware to simplify maintenance and fault diagnosis.

The primary computing system uses a Jetson Orin AGX [2], while navigation data are obtained through a custom PCB integrating an IMU and an external marine-grade Garmin GPS. This removes dependence on the previously used Pixhawk architecture and reduces integration complexity.

Hardware and software safety systems monitor battery state, communications, and system faults. Emergency-stop circuitry provides direct propulsion shutdown independent of the autonomy software.

C. USV: Perception System

The USV perception architecture combines a ZED 2i stereo camera with a Velodyne VLP-16 Hi-Res LiDAR. Camera-based semantic detections provide object classification, while LiDAR clustering provides geometric localization and obstacle information. Both are fused to improve detection reliability and situational awareness. Further details of the perception algorithms and semantic mapping framework are provided in Appendix F.

D. USV: Autonomy Software

The autonomy stack is implemented in ROS 2 and separates mission management, perception, planning, and control. An Artificial Potential Field (APF) planner generates collision-aware trajectories using onboard perception and information shared by the UAV. Low-level control is performed by a Deep Reinforcement Learning (DRL) policy trained using Proximal Policy Optimization (PPO) [3], with commands transmitted through a Teensy 4.1 [4] to the T500 thrusters.

This modular architecture enables the same autonomy components to support multiple RobotX tasks. Further details of the DRL controller are provided in Appendix C.

The APF planner was evaluated using competition-style buoy configurations to verify semantic obstacle avoidance and correct passing behavior. Figure 5 shows the USV trajectory and the corresponding on-water testing environment.

E. UAV Design: Early Stages

The unmanned aerial vehicle (UAV) early design stages included many phases of research to ensure that the tasks required for the RobotX competition could be successfully completed. Funding restrictions required that the UAV and its components were also compliant with the National Defense Authorization Act (NDAA). The team selected a main computer, flight controller, sensors, control link protocol,

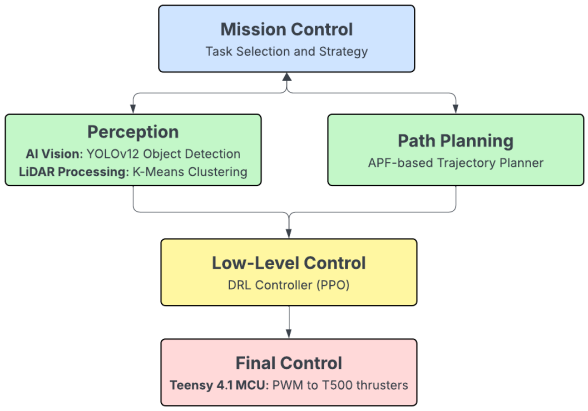


Fig. 4. High-level USV autonomy architecture.

batteries, and radios capable of performing the tasks required for the competition. Trade studies were completed to compare different options before starting the design of the custom built UAV as shown in appendix O. Figure 6 shows the CAD for the custom built UAV design.

Prototype parts for the frame were 3d printed in Polylactic Acid (PLA) plastic and used for initial testing, as shown in 7. 500 mm Carbon fiber tubes with 25 mm diameter were used for motor arms with custom electronic speed controller (ESC) and motor mounts.

Due to the complexity of the custom build, the ModalAI Starling 2 Max model C29 was developed in parallel to the custom built UAV using the same autonomy software stack, allowing for parallel test and evaluation.Due to time constraints and development obstacles, the custom UAV will not be used for the 2026 RobotX competition, and the Starling 2 Max will serve as the UAV. The full modification logs for the Starling 2 Max used to prepare for the competition are shown in Appendix U.

F. UAV Design: Frame Design

The ModalAI Starling 2 Max is a commercial off the shelf (COTS) development drone that can be used for GPS-denied navigation and sensing/reconnaisane tasks using computer vision. The baseline C29 model has a take-off weight of 566 g and a payload capacity of around 930 g. The frame is composed of 3 mm carbon fiber that mounts four 2203.5-size, 1500-KV brushless motors with 180 mm Tri-Blade propellers. The ModalAI 4-in-1 Mini ESC is mounted below the main computer. Batteries are mounted beside the center frame [5].

The base model was originally modified to mount an additional radio for telemetry. Manufacturer provided CAD models were used to design a custom radio mount that attached to the Starling 2 Max using existing screws that attached to the carrying handle and the forward facing camera mount while clearing any downward facing sensors.

Due to changes in Singapore law restricting the use of the RFD900ux radio in late 2025 [6], this design was replaced with an 2.4 GHz Express Long Range System (ELRS) Module, and the additional mount was removed. This was replaced with mounts to be used for a robotic gripper. The mount was designed so that the gripper could be easily added or removed using an M5 bolt. The mount design can be observed from the bottom of the UAV as shown in Fig. 8a, and the front view of the UAV with the robotic gripper attached is shown in Fig. 8b. The full robotic gripper design is discussed in appendix I.

The landing gear was modified so that the UAV would float in case of an emergency landing. The original landing gear was completely removed and replaced with Thermoplastic Polyurethane (TPU) posts with 3 inch circular openings at the bottom to mount two polyethylene foam cylindrical flotation devices in parallel, as depicted in Fig. 8. The flotation mechanism was placed directly below the motors to minimize the effect on the lift of the UAV. The use of TPU for the flotation mounts also provided impact resistance and safer landing.

G. UAV Design: Electrical Design

The Starling 2 Max is powered via the VOXL Mini 4-in-1 ESC which provide built-in power management while also supplying power to the motors. Two 2S 18650 Li-Ion batteries are connected in series via XT30 connectors, providing a total max voltage of 16.8V [5]. A custom harness was used to split the 5V power provided to the VOXL2 in order to power the additional radio. The wiring diagram provided by ModalAI was modified to reflect this change as shown in appendix R.

The C29 model of the ModalAI starling 2 max is already equipped with dual Sony IMX412 12MP RGB cameras, dual AR0144 fisheye CV 1MP tracking cameras, and a time-of-flight (ToF) sensor. The wiring for the vision system is shown in Appendix S.

H. UAV Design: Communication Links

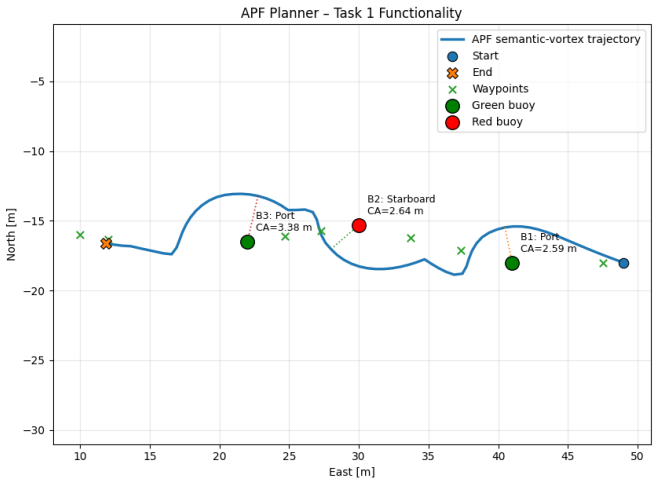
The UAV uses three types of communication links for: 1) remote control (RC); 2) telemetry; and 3) multiagent coordination over a wireless local area network (WLAN). These systems allow bi-directional communication between the UAV and the operator control station (OCS), which then communicates with the USV OCS, as depicted in Fig. 9.

1) Radio Control

The RC link is used for manual remote control of the UAV. This link is required for safe operation of the UAV during both manual and autonomous missions. It is obtained using n RP1 V2 ExpressLRS 2.4 GHz Nano Receiver [7] on the UAV connected to the PX4 flight controller on the UAV, published on the $input_rc$ and $rc_channels$ ROS2 topics. This is bound to a Jumper AION ELRS 2.4 GHz transmitter module in a RadioMaster Pocket controller [8].

2) Telemetry

The initial design included an RFD900ux radio on both the UAV and ground control station. Due to Singapore regulations restricting UAV communication in 900 MHz frequency bands, an alternative 2.4 GHz frequency system was acquired. An additional RP1 V2 ExpressLRS 2.4 GHz Nano Receiver and Jumper AION ELRS 2.4 GHz transmitter module were



(a) APF trajectory demonstrating correct avoidance of red and green buoys.



(b) USV during on-water testing with competition-style buoys.

Fig. 5. APF-based navigation testing for RobotX Task 1 functionality: (a) planned USV trajectory and (b) physical testing environment.

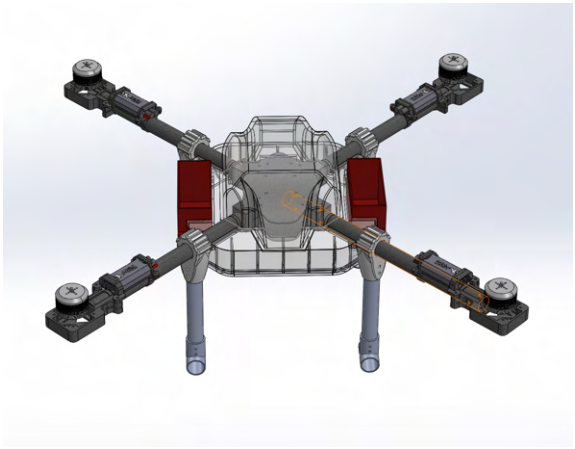


Fig. 6. CAD for custom UAV design.



Fig. 7. Custom UAV design prototype fully assembled.

implemented in MAVLink mode to establish the telemetry link between the UAV and QGroundControl.

3) Wireless Local Area Network

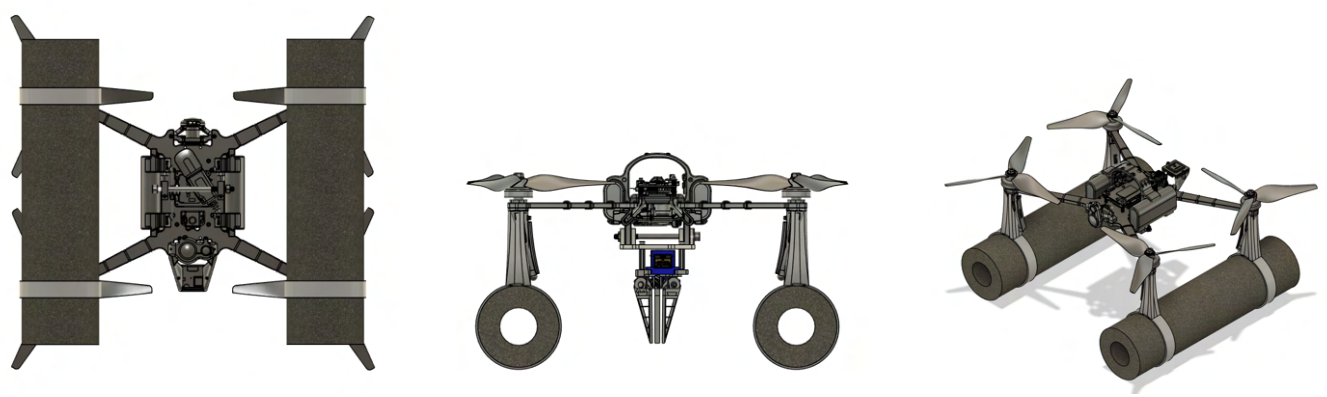
An ALFA Network AWUS036EACS dual-band USB wireless adapter is connected via the M0151_A UART adapter board on the VOXL2 computer. This is used to establish Wi-Fi connectivity or set up a wireless access point supporting both 2.4 GHz and 5 GHz communication [9]. The UAV will be configured to use the same 2.4 GHz access point as the USV during the competition.

I. UAV Design: Autonomy Software Stack

The main computer onboard the UAV is a VOXL2 which contains an onboard neural network accelerator, optimizing computer vision algorithms that require object detection [5].

The software stack uses the robot operating system 2 (ROS2) which takes input from MAVROS and PX4. MAVROS is a MAVLink communication node for ROS2 [10]. MAVLink is a messaging protocol used for drone communication. It includes data streams published as topics and sub-protocols for mission and parameter control [11]. ROS2 is used to handle navigation and sensing algorithms. Specifically machine learning is used to detect and locate objects on the water surface for our missions. Figure 10 shows the software stack that is used on-board the UAV. The voxl architecture diagram is shown in Appendix P.

A modified version of the Rapidly-Exploring Random Tree (RRT) algorithm was used for path planning and autonomous exploration. Assuming an arbitrary take-off location is used, the algorithm instructs the UAV to search and explore the area-of-interest and uses detected objects to assign rewards when planning the best path. Beacon detections from the algorithm



(a) Bottom view. (b) Front view with robotic gripper integrated. (c) Isometric view with landing gear.

Fig. 8. Robot-X 2026 modified Starling 2 Max CAD design.

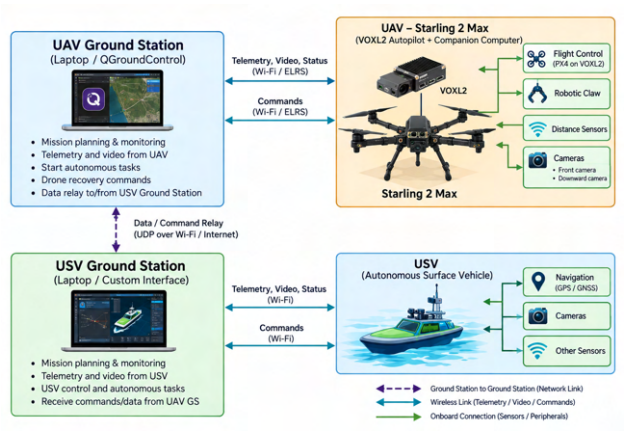


Fig. 9. UAV-USV wireless communications coordination architecture.

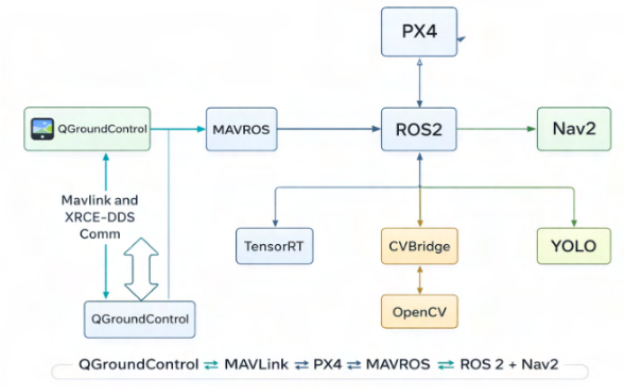


Fig. 10. UAV autonomy software stack.

described in Appendix H guide the exploration.

Autonomous scripts are triggered by a system service programmed to listen to channel 7 on the radio controller. When channel 7 is high, it will create a fastdds profile and start the voxl-mpa-ros2 service that converts the voxl modal pipe architecture (MPA) to ROS2 messages [12]. It also triggers

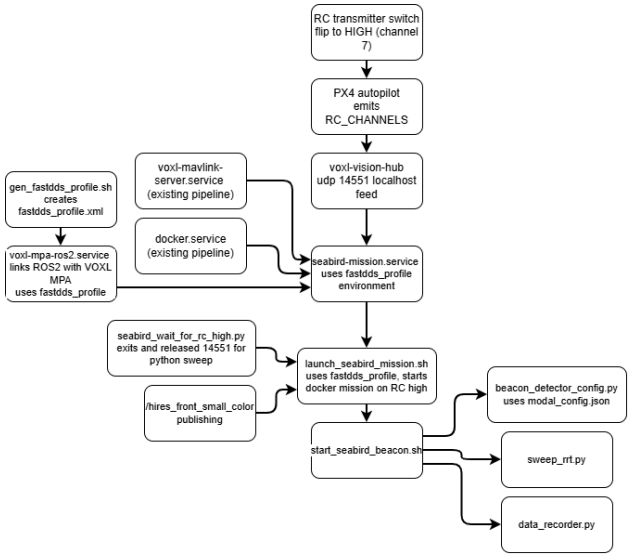


Fig. 11. UAV beacon detection architecture.

a service that starts a docker container with ROS2 Humble and other dependencies required for the beacon detection algorithm. Images from the `hires_front_small_color` topic are fed into the beacon detection algorithm described in Appendix H. The architecture for the autonomous beacon detection algorithm is shown in Fig. 11.

J. UAV Design: Safety Protocols

Several safety protocols and fail-safe procedures are used to ensure the safe operation of the UAV for each flight. For each flight, a certified part 107 pilot is present and serves as remote pilot in command (RPIC). Each of the following are essential elements of the UAV safety protocols.

1) *Preflight Checklist*: A preflight checklist is used at the start of each flight to ensure that the drone is clear to fly safely and legally. Local restrictions related to regulation and weather conditions are verified as well as physical

components of the drone. The full preflight checklist is available in Appendix T.

2) *Visual Line of Sight (VLOS)*: Throughout each mission, the RPIC maintains VLOS of the drone for the duration of the mission. This helps maintain RC and telemetry links and ensures that the drone stays clear of obstacles.
3) *Emergency Stop*: Emergency stop procedures are configured on the drone in the event that it is unsafe to fly and power must be cut to the motors. This is the final contingency if the drone is unsafe during flight, but it is also used when the drone is not in use to prevent the drone from arming accidentally.
4) *Return-to-Launch (RTL)*: The RTL function is called by several other fail-safe procedures in the event that the drone cannot operate safely, but implementing an emergency stop on the motors is not yet necessary. In the event that the UAV must return to launch, it will hover at the takeoff location for 10 seconds at 65 ft altitude before landing. This additional time is alotted to ensure the area below the UAV is clear of obstacles before it lands.
5) *Loss of Manual Control*: Manual control is essential for safe operation of the UAV. Even if the UAV is operating autonomously, manual control must remain available for emergency recoveries. If the manual control link is lost for more than 1 second, the UAV will initiate RTL protocol.
6) *Loss of Telemetry*: While telemetry is important for operating missions safely, diligent effort allows safe operation even when the telemetry link is lost temporarily. Given that a licensed RPIC is available, the UAV will continue missions even if the telemetry link is lost. The control station will provide both visual and audible notifications if the telemetry link is lost for 20 seconds.
7) *GPS/GNSS Loss*: A global positioning system (GPS) or Global Navigation Satellite System (GNSS) is required when the UAV is operating in position and mission flight mode, however it may still operate safely in altitude flight mode when there is no reliable GPS available. In the event that GPS signal is lost while the UAV is in position mode, it will switch to altitude mode. If a valid height estimate cannot be determined, the UAV will initiate RTL protocol.
8) Battery Failsafe: The UAV battery state of charge (SoC) is consistently monitored using the built in battery management system and notifications in QGroundControl. To ensure the UAV can fly and return home safely, battery SoC thresholds are set to prevent loss of power during flight. The following procedures are configured for each SoC threshold:

- 15% SoC - Visual and Audio low battery warning
- 10% SoC - RTL protocols initiated
- 5% SoC - Emergency landing at current location
- 10% SoC before flight - the UAV will not arm

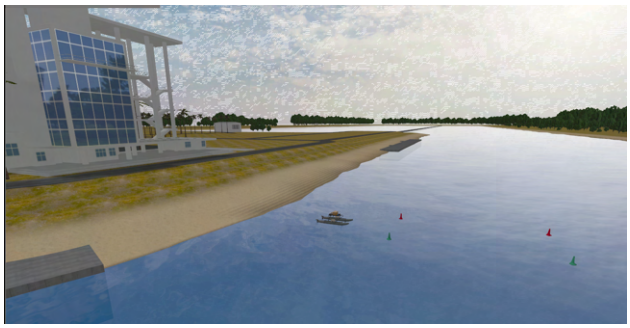


Fig. 12. ASV attempting Task 1 in the VRX simulator.

IV. Testing Strategy

Based on experience from previous RobotX and RobotBoat competitions, the team's testing strategy emphasizes early software validation, progressive system integration, and frequent iteration without constraining hardware development. To achieve this, testing is conducted across three complementary layers: simulation, offline data-driven validation, and on-water testing. This layered approach enables rapid development of autonomy algorithms while progressively increasing environmental realism and system complexity.

A. Simulation

Simulation is the primary environment for early development and regression testing of the autonomy stack. The team uses the Virtual RobotX (VRX) Gazebo simulator and IsaacSIM to develop and validate mission logic, perception pipelines, and APF-based path-planning algorithms under controlled conditions. Custom simulation worlds are created to approximate RobotX tasks, including navigation channels, debris fields, and docking scenarios. Simulation enables the rapid iteration of mission logic and control behaviors, independent of hardware availability, while providing a consistent framework for testing failure cases and edge conditions.

B. Data Sharing Testing

Offline testing using shared datasets is critical for validating perception and planning algorithms. The team leverages the RoboNation Data Sharing platform and archived datasets from previous RobotX and RobotX competitions to train and evaluate YOLO-based object detection models and LiDAR clustering algorithms. Recorded ROS bag files containing camera and point cloud data are used to verify detection robustness, classification accuracy, and planner behavior prior to deployment on the vehicle. This approach allows perception performance to be evaluated across a wide range of environmental conditions without requiring repeated field testing.

C. On-Water Testing

On-water testing is conducted to validate full system integration and assess ASV performance under real-world conditions, including wind, current, and waves. Testing progresses from basic navigation and obstacle avoidance to task-level

execution, evaluating perception robustness, APF-based planning, and the deep reinforcement learning control policy under unmodeled dynamics. In addition to RobotX-specific testing, the team' s research group has previously exercised core autonomy components on a WAM-V 16 platform, providing added confidence in the transferability of the underlying algorithms. The UAV was first tested on land before testing while flying over bodies of water.

ACKNOWLEDGMENT

Team SeaOwls gratefully acknowledges the support of RoboNation, FAU's College of Mechanical and Ocean Engineering, FAU's Center for Connected Autonomy and Artificial Intelligence, FAU Student Government, and The Link Foundation for their financial support of this project. We also thank our faculty advisors, Dr. Manhar Dhanak, Dr. Pierre-Philippe Beaujean, Dr. Georgios Sklivanitis, and Dr. Dimitris A. Pados for their continued technical guidance and mentorship, which have been instrumental in the development, integration, and testing of the Autonomous Surface Vehicle and Autonomous Aerial Vehicle. Their expertise and support have played a critical role in advancing the team's research and competition efforts.

REFERENCES

[1] X. Vicent, R. Velasco, E. Diaz, C. Wilson, and M. R. Dhanak, "Navigating dynamic waters: Learning-based control and potential-field planning for asvs," in *Proceedings of RobotBoat 2026*, Sarasota, FL, USA, Mar. 2026.

[2] NVIDIA, "NVIDIA Jetson AGX Orin," [Online]. Available: https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-orin/, accessed: Oct. 6, 2024.

[3] L. Engstrom, A. Ilyas, S. Santurkar, D. Tsipras, F. Janoos, L. Rudolph, and A. Madry, "Implementation matters in deep policy gradients: A case study on PPO and TRPO," *arXiv preprint arXiv:2005.12729*, 2020.

[4] PJRC, "Teensy 4.1," [Online]. Available: https://www.pjrc.com/store/teensy41.html, accessed: Oct. 6, 2024.

[5] ModalAI, Inc., "Starling 2 max," *ModalAI*. [Online]. Available: https://www.modalai.com/products/starling-2-max, accessed: Sep. 17, 2026.

[6] Civil Aviation Authority of Singapore, "Broadcast remote identification requirement for unmanned aircraft to come into effect from 1 december 2025," [Online]. Available: https://www.caas.gov.sg/resources/media-and-publication/newsroom/broadcast-remote-identification-requirement-for-unmanned-aircraft-to-come-into-e/, Oct. 28 2025, accessed: Sep. 18, 2026.

[7] RadioMaster, "RP1 V2 ExpressLRS 2.4 GHz Nano Receiver," Product webpage. [Online]. Available: https://radiomasterrc.com/products/rp1-expresslrs-2-4ghz-nano-receiver, accessed: Sep. 18, 2026.

[8] Jumper Technology, "Jumper 2.4G ELRS TX Module," Product webpage. [Online]. Available: https://www.jumper-rc.com/modules/jumper-2-4g-elrs-tx-module/, accessed: Sep. 18, 2026.

[9] ALFA Network Europe, "ALFA Network AWUS036EACS—802.11ac Wi-Fi and Bluetooth USB Adapter," Product webpage. [Online]. Available: https://alfa-network.eu/alfa-usb-adapter-awus036eacs, accessed: Sep. 18, 2026.

[10] MAVROS Contributors, "MAVROS 2.9.0 Documentation—ROS 2 Iron," [Online]. Available: https://docs.ros.org/en/iron/p/mavros/, accessed: Sep. 17, 2026.

[11] MAVLink Project, "MAVLink Developer Guide," [Online]. Available: https://mavlink.io/en/, accessed: Sep. 17, 2026.

[12] ModalAI, "ROS 2 Installation on VOXL 2," *ModalAI Technical Documentation*. [Online]. Available: https://docs.modalai.com/ros2-installation-voxl2/, Jul. 2 2026, accessed: Sep. 18, 2026.

[13] ModalAI, Inc., "VOXL SDK," *ModalAI Technical Docs*. [Online]. Available: https://docs.modalai.com/voxl-sdk/, 2026, accessed: Sep. 16, 2026.

[14] ——, "VOXL 2 D0014 Architecture (Starling 2)," *ModalAI Technical Docs*. [Online]. Available: https://docs.modalai.com/voxl2-d0014/, Sep. 3 2026, accessed: Sep. 16, 2026.

APPENDIX A
HULL DESIGN

The RobotX 2026 Autonomous Surface Vehicle utilizes an outboard asymmetrical catamaran hull to address buoyancy, stability, and serviceability limitations identified in the previous platform. The prior hull operated with insufficient freeboard at an overall vehicle mass of approximately 110 lb, increasing susceptibility to water ingress and reducing reliability during competition. The redesigned hull increases displacement margin while remaining compliant with RobotX dimensional constraints, providing improved freeboard at full system weight.

The asymmetrical catamaran geometry features a curved outboard hull surface and a flat vertical inboard face, which shifts the centers of buoyancy outward and increases transverse stability (Fig. A.1). This configuration reduces hydrodynamic interference between hulls and decreases wetted surface area, improving propulsion efficiency and low-speed maneuverability. The hull was designed with a waterline length of 1.83 m; assuming a conservative Froude number of 1.1, the corresponding displacement velocity is 4.66 m/s, which is attainable given the available propulsion thrust.

Hull geometry was developed using a set of cross-sectional stations lofted in SolidWorks along guide curves aligned with the deck, keel, and design waterline. Structural aluminum cross-beams connect the hulls and provide rigid, accessible mounting interfaces for the deck and central electronics enclosure, improving structural integrity and maintainability.

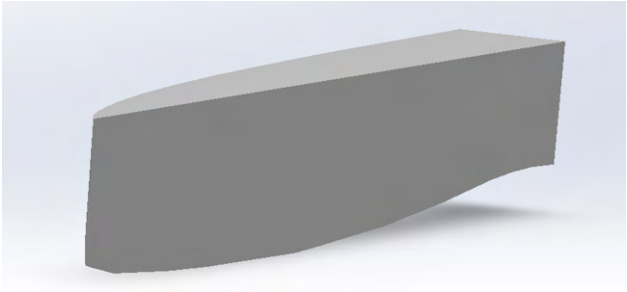


Fig. A.1. CAD of the new hull.

TABLE A.1
HULL GEOMETRY AND DESIGN RATIOS

Length	5.92 ft
Length/Beam	9.4
Beam (hull)	0.63 ft
Beam (total)	1.77 ft
Separation Ratio	0.2
Height	0.86 ft
Waterline	0.33ft
Volume (1 hull)	1.63 cubic feet

TABLE B.1
PERFORMANCE RESULTS OF SYSTEM IDENTIFICATION: BASELINE AND CMA-ES

Metrics	Baseline	CMA-ES
Acceleration RMSE (u) m/s	0.232	0.072
Acceleration RMSE (v) m/s	0.068	0.066
Acceleration RMSE (r) rad/s	0.013	0.014
Circle RMSE (u) m/s	0.317	0.099
Circle RMSE (v) m/s	0.097	0.097
Circle RMSE (r) rad/s	0.153	0.019

APPENDIX B
ASV HYDRODYNAMIC MODELING AND PARAMETER OPTIMIZATION

This Appendix describes the dynamic modeling and system identification process used to obtain a high-fidelity three-degree-of-freedom (3-DOF) planar model of the ASV. Vehicle dynamics are formulated in a body-fixed reference frame, capturing surge, sway, and yaw motion.

The nonlinear equations of motion are expressed as:

$$\mathbf{M}_b\dot{v} + \mathbf{C}_b\left(v\right)v + \mathbf{D}_b\left(v\right)v = \tau_b$$

where

$$v = \begin{bmatrix} u & v & r \end{bmatrix}^T$$

denotes the body-frame velocities, and

$$\tau_b = \begin{bmatrix} F_x & F_y & M_z \end{bmatrix}^T$$

represents the control forces and yaw moment. The inertia, Coriolis, and damping matrices incorporate both rigid-body and hydrodynamic added-mass effects, in accordance with standard formulations for marine vehicle modeling.

Initial hydrodynamic coefficients are obtained from benchmark parameterizations for WAM-V–class vessels and refined using experimental data collected from acceleration, circle, and zig-zag maneuvers performed on the physical ASV. These tests capture surge drag, coupled sway–yaw dynamics, and turning behavior.

Parameter refinement is performed using a Covariance Matrix Adaptation Evolution Strategy (CMA-ES). For each candidate parameter set, the simulated ASV response is compared against measured data using identical thrust inputs. The optimization objective minimizes the weighted root mean square error (RMSE):

$$J = \mathrm{RMSE}\left(u\right) + \mathrm{RMSE}\left(v\right) + \lambda\mathrm{RMSE}\left(r\right)$$

where an increased weighting λ is applied to the yaw-rate error to prioritize accurate turning dynamics.

The resulting optimized model closely matches the physical ASV response and serves as the simulation environment for training and evaluating deep reinforcement learning–based controllers.

TABLE C.1
PERFORMANCE RESULTS OF PID AND DRL CONTROLLERS

Controller	Mean Speed (ms)	Heading Rate RMS (rad/s)	Mean WP error (m)	CRE RMS (m)
PID HS	0.99	0.062	1.00	1.05
DRL	1.03	0.068	0.76	0.97

The CMA-ES optimization significantly improves the accuracy of the ASV model compared to the benchmark parameters. In the acceleration tests, surge RMSE is reduced from 0.232 m/s to 0.072 m/s, indicating a much better representation of longitudinal dynamics. In the circle tests, yaw-rate RMSE is reduced from 0.153 rad/s to 0.019 rad/s (≈88% reduction), demonstrating a substantial improvement in turning behavior. These results confirm that the optimized model more accurately captures both straight-line and maneuvering dynamics, making it suitable for controller design and DRL training.

APPENDIX C
DEEP REINFORCEMENT LEARNING CONTROLLER FOR ASVS

This Appendix details the deep reinforcement learning (DRL) controller used for low-level ASV navigation and trajectory tracking. The controller is trained using Proximal Policy Optimization (PPO) on the CMA-ES–optimized hydrodynamic model described in Appendix B.

The observation space is defined using line-of-sight navigation variables, cross-track error, heading alignment, and body-frame velocities, enabling the policy to generalize across varying initial conditions without reliance on predefined paths. The action space consists of normalized differential thrust commands applied to the port and starboard thrusters.

The reward function is designed to promote efficient waypoint convergence, heading alignment, and stable motion while penalizing lateral drift and excessive yaw rate:

$$R_t = k_d\left(d_{t-1} - d_t\right) + k_\psi\cos\left(\psi_{\mathrm{LOS}}\right) - k_v\left|v_t\right| - k_\omega\left|\omega_t\right| + k_{\mathrm{time}}$$

where d_t is the distance to the waypoint, ψ_{LOS} is the line-of-sight heading error, v_t is lateral velocity, and w_t is yaw rate. A terminal bonus is applied upon reaching the waypoint within a specified tolerance.

Training episodes consist of waypoint navigation tasks with randomized initial positions and headings, and waypoint distances ranging from 3 to 10 meters.

Controller performance is evaluated in simulation and compared against a baseline PID heading–speed controller. As shown in Table C.1, the DRL controller achieves lower mean waypoint error (0.76 m vs. 1.00 m) and reduced cross-track error RMS (0.97 m vs. 1.05 m), while maintaining comparable speed and heading-rate stability. These results demonstrate improved path-following accuracy and robustness, validating the effectiveness of combining learning-based control with a high-fidelity, data-driven vehicle model.

APPENDIX D
PERCEPTION ALGORITHMS: LIDAR CLUSTERING AND VISION-BASED OBJECT DETECTION

K-means clustering is employed to segment LiDAR point cloud data into distinct object candidates for obstacle detection and localization. The algorithm partitions a dataset into k clusters by minimizing the intra-cluster variance. The set of cluster centroids is defined as

$$C = \{c_1, c_2, \ldots, c_k\}. \tag{D.1}$$

Each data point x_j is assigned to the nearest centroid by Euclidean distance:

$$S_i = \{x_j : \|x_j - c_i\| \leq \|x_j - c_m\|, \quad \forall m \in \{1, \ldots, k\}\}. \tag{D.2}$$

Centroids are updated by averaging points assigned to each cluster:

$$c_i = \frac{1}{|S_i|} \sum_{x_j \in S_i} x_j. \tag{D.3}$$

The clustering process minimizes the objective function

$$J = \sum_{i=1}^{k} \sum_{x_j \in S_i} \|x_j - c_i\|^2 \tag{D.4}$$

and iterates until convergence, defined by

$$\|c_i(t+1) - c_i(t)\| < \epsilon, \quad \forall i \in \{1, \ldots, k\}. \tag{D.5}$$

After convergence, clusters are filtered using geometric constraints such as size, height, and spatial consistency to identify competition-relevant objects, including buoys and obstacles.

To complement LiDAR-based perception, the ASV employs a deep-learning vision pipeline based on the *You Only Look Once* (YOLO) object detection framework. YOLO performs single-stage, real-time object detection by processing the entire image in one forward pass, enabling high inference speed while maintaining accurate localization and classification.

The model is fine-tuned using a custom maritime dataset containing competition-specific objects, including navigation buoys, docking structures, and vessels, captured under varying lighting conditions and water surface reflections. This training improves robustness in low-contrast and low-visibility environments commonly encountered during RobotX operations.

The optimized YOLO model is deployed on NVIDIA Jetson-class embedded processors, enabling efficient, real-time inference within the ASV's perception pipeline. Detected objects are published as ROS 2 messages and fused with LiDAR-based clustering outputs to provide semantic classification and geometric localization. This combined perception approach enhances detection reliability and situational awareness across navigation and task execution scenarios.

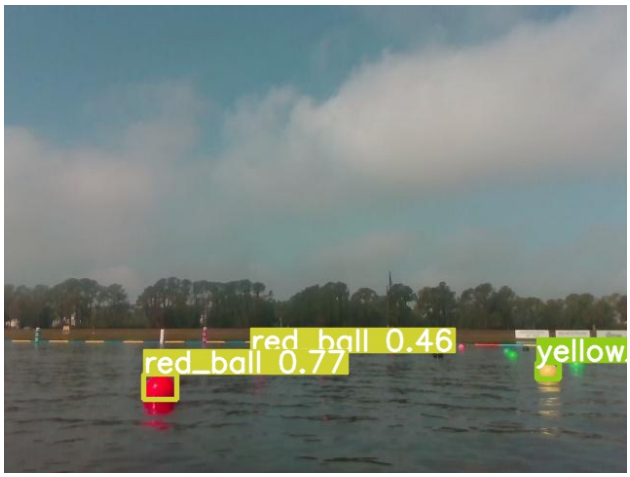


Fig. D.1. YOLO26 custom model detector output.

APPENDIX E
USV TESTING PLAN

A. Scope

Team SeaOwls employs a layered testing strategy to validate the Autonomous Surface Vehicle (ASV) across perception, planning, and control subsystems while minimizing dependency on hardware availability. Testing is structured to progress from simulation and offline data-driven validation to full system integration and on-water deployment. The Virtual RobotX (VRX) simulator and RoboNation Data Sharing resources form the foundation of early development, enabling rapid iteration prior to physical testing.

B. Schedule

The testing schedule is organized around three primary phases: simulation-based development, offline validation using recorded datasets, and on-water system integration. Table E.1 summarizes the testing timeline, objectives, and environments.

Integrated Mission Testing Via Gazebo

C. Resource & Tools

Testing activities leverage a combination of software and physical resources. Simulation and offline testing are conducted using the VRX Gazebo simulator on dedicated desktop workstations. Perception development relies on datasets from the RoboNation Data Sharing platform and archived ROS bag files. Hardware development and integration are supported by FAU's electronics and machine shop facilities.

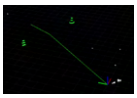
For on-water testing, the ASV is deployed using a davit system, and anchored buoys are used to replicate elements of the competition task. These resources allow progressive validation of perception, planning, and control subsystems in increasingly realistic environments.

D. Environment

Software testing is initially conducted in simulation and using shared datasets due to limited access to competition-grade physical objects. This approach enables validation of perception algorithms and mission logic prior to physical deployment.

On-water testing is conducted at FAU's SeaTech facility along the US Intracoastal Waterway in Dania Beach, Florida. Testing is performed within a protected marina environment to minimize external traffic and environmental risks while enabling the controlled evaluation of full system autonomy.

TABLE E.1
TEAM SEAOWLS USV TESTING SCHEDULE AND RESULTS

Start/End Date	Documentation	Target	Result	Environment	Member Presence
03/20/26 - 04-31-26		YOLO vision model training using shared datasets	$\geq$ 0.60 confidence in competition objects	RoboNation Data Sharing, RoboFlow, MRC Server	AI Team
08/20/25 - 09/15/25		LiDAR clustering and object detection validation	Reliable detection of buoys and docking bays	Shared ROS bags, RViz	Software & Controls Team
05/01/26 - 07/31/26		Mission logic and APF path planning validation	Task completion without vision dependency	VRX Simulator	Software & Controls Team
05/31/26 - 07/31/26		DRL controller training and evaluation	Stable waypoint tracking (3–10 m)	Simulation	Software & Controls Team
08/01/26 – 08/31/26		Integrated mission testing;	End-to-end autonomy validation	VRX Simulator	Full USV Team
08/20/26 - 10/31/26		On-water system testing	Full system integration under disturbances	On-Water tests at the US-Intracoastal Waterway in Dania Beach, FL	Full USV Team;
09/20/26 - 10/31/26		On-water system testing with UAV	System of Systems integration under disturbances	On-Water tests at the US-Intracoastal Waterway in Dania Beach, FL	Full Team;

E. Risk Management

Safety considerations are integrated into all testing activities. Battery handling and charging procedures follow established protocols to mitigate risks associated with lithium-polymer batteries. All work conducted in the electronics and machine shops adheres strictly to FAU laboratory safety guidelines.

On-water testing introduces additional risks due to wildlife, vessel traffic, and deployment operations. To mitigate these risks, testing sessions are coordinated with FAU SeaTech staff, required documentation is completed in advance, and all team members receive a safety briefing prior to deployment. Personal protective equipment, including hard hats, is mandatory during davit operations.

F. Results

Between Spring, Summer, and Fall 2026, the team validated key components of the autonomy stack through simulation, shared datasets, and limited physical testing. Notable outcomes include stable waypoint tracking with the DRL controller, successful task execution without vision-based navigation in simulation, and verification of the electronics enclosure' s waterproofing. The use of shared RoboNation datasets significantly accelerated perception development and reduced reliance on early field testing.

APPENDIX F
USV SEMANTIC-MAPPING PAPER

AI-Assisted Real-Time Vision–LiDAR Semantic Mapping of Maritime Obstacles for Autonomous USV Navigation

Xavier Vicent
Dept. of Ocean & Mechanical Engineering
Florida Atlantic University
Dania Beach, FL, USA
xvicentnavar2024@fau.edu

Daniel Morris
Dept. of Ocean & Mechanical Engineering
Florida Atlantic University
Dania Beach, FL, USA
danielmorris2022@fau.edu

Manhar R. Dhanak
Dept. of Ocean & Mechanical Engineering
Florida Atlantic University
Dania Beach, FL, USA
dhanak@fau.edu

Abstract— Reliable obstacle perception and mapping are essential for autonomous Unmanned Surface Vehicle (USV) navigation. This paper presents an AI-assisted real-time semantic mapping framework combining stereo vision, LiDAR sensing, machine learning, and dynamic Kalman filtering.

A stereo camera with a custom YOLOv8 detector provides semantic buoy detections and stereo localization, while a 360-degree LiDAR provides geometric obstacle measurements. Camera and LiDAR observations are fused and processed using XGBoost models for measurement reliability, position correction, and uncertainty estimation, followed by a Random Forest-assisted dynamic Kalman filter for persistent semantic mapping. Field experiments showed that the camera-only AI-assisted mapper reduced mean buoy localization error from 1.122 m to 0.882 m and prevented duplicate landmark formation in a close-buoy experiment. Physical camera–LiDAR tests reduced held-out fused-measurement error from 0.813 m to 0.050 m. In a 20-buoy Virtual RobotX experiment, the complete framework reduced mean semantic-map localization error from 0.550 m to 0.123 m while mapping all visible landmarks.

Index Terms— Unmanned Surface Vehicles, Semantic Mapping, Camera–LiDAR Fusion, Kalman Filtering, Machine Learning, Maritime Robotics

A. Introduction

Unmanned Surface Vehicles (USVs) are increasingly employed in maritime applications including bathymetry mapping [1], environmental monitoring [2], subsea magnetic and electric field mapping [3], and autonomous navigation and control [4]–[8]. These missions require perception systems capable of detecting surrounding objects and maintaining persistent representations for autonomous navigation. Camera-based methods provide semantic information [9], [10], while stereo vision additionally provides three-dimensional localization.

In this work, a ZED 2i stereo camera and custom YOLOv8 detector [11] identify and localize red and green navigation buoys. Stereo measurements are transformed into a global North-East-Down (NED) frame using the USV position and heading. However, reflections, stereo-depth uncertainty, vehicle-pose error, sensor timing, and repeated observations can degrade association and generate duplicate landmarks. A dynamic Kalman filter (KF) can maintain persistent landmarks [12], but conventional association thresholds and fixed measurement covariance do not adapt to changing observation quality. Random Forest models [13] are therefore introduced to estimate detection-to-track reliability and measurement uncertainty using self-supervised KF-track consistency.

Previous maritime multi-sensor approaches have primarily focused on obstacle-level fusion, detection, or tracking. Liu et al. [14] used Dempster–Shafer evidence theory to construct an obstacle grid map, while Ponzini and Martelli [15] used camera–LiDAR fusion for maritime classification and tracking. In contrast, the proposed framework maintains persistent semantic landmarks in a global frame and applies learned reliability, position correction, and uncertainty estimation at both the sensor-fusion and dynamic mapping stages.

The framework incorporates a Velodyne VLP-16 Hi-Res LiDAR, geometric camera–LiDAR association, XGBoost models [16] for measurement reliability, position correction, and uncertainty estimation, and Random Forest-assisted dynamic KF mapping. Evaluation includes camera-only and camera–LiDAR physical field experiments and a 20-buoy Virtual RobotX (VRX) experiment [17].

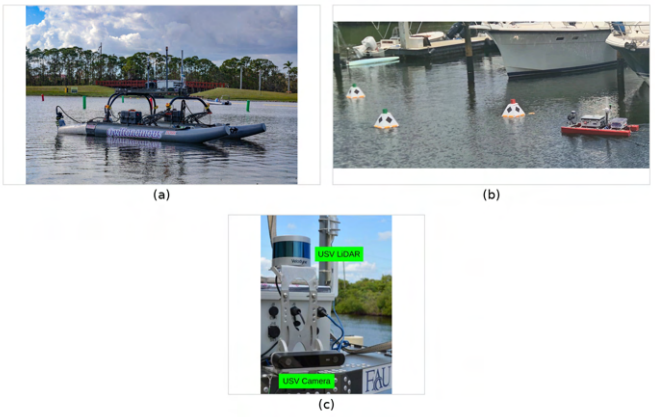


Fig. F.1. Experimental platforms and perception system: (a) WAM-V USV 16, (b) custom 1.8-m ASV, and (c) ZED 2i stereo camera with Velodyne VLP-16 LiDAR.

B. Methodology

This section describes the proposed semantic mapping framework. The system combines stereo-camera object detection, LiDAR obstacle measurements, vehicle-pose information, tree-based learning models, and dynamic Kalman filtering to generate a persistent semantic obstacle map in the global NED frame.

1) Experimental Platforms and Perception System

Two physical USVs were used during development and field evaluation. The first was a custom 1.8-m asymmetrical catamaran ASV with a mass of 49.3 kg and two stern-mounted thrusters, developed as part of Team Owltonomous' RoboBoat 2026 platform [18]. The second was the 4.88-m, approximately 180-kg WAM-V USV 16 [19]. The experimental platforms and perception sensors are shown in Fig. F.1.

A common Guidance, Navigation, and Control (GNC) enclosure was transferred between the platforms. An NVIDIA Jetson AGX Orin 32 GB executed the ROS 2 perception, fusion, and mapping nodes. Vehicle position and heading were obtained from an SBG IG500N navigation sensor. The perception system consisted of a ZED 2i stereo camera and a Velodyne VLP-16 Hi-Res LiDAR.

2) Stereo-Vision Object Detection and Localization

A custom YOLOv8 model [11] processes the left ZED 2i image to detect red and green navigation buoys. The resulting two-dimensional detections, including semantic class, confidence, and bounding box, are passed to the Stereolabs ZED SDK [20], where stereo depth is used to estimate the three-dimensional position of each detected buoy relative to the camera.

After applying the calibrated camera-to-body transformation, the measurement is projected into the global NED frame as

$$\mathbf{p}_i^{NED} = \mathbf{p}_v^{NED} + \mathbf{R}(\psi_v)\mathbf{p}_i^b, \tag{F.1}$$

where $\mathbf{p}_v^{NED}$ and ψ_v are the USV position and heading, and $\mathbf{p}_i^b$ is the detected buoy position in the body frame. The resulting semantic observation is supplied either directly to the mapper or to the camera–LiDAR fusion stage.

3) LiDAR Obstacle Detection

The VLP-16 point cloud is filtered to remove invalid measurements and segmented using PCL Euclidean clustering with a KD-tree search structure [21]. For each cluster, the centroid and three-dimensional bounding box are calculated. Geometric filtering based on cluster dimensions, point count, height, and range is then applied to reject objects inconsistent with buoy-like obstacles. The centroid of each accepted cluster is used as the LiDAR-derived obstacle position.

4) Camera–LiDAR Fusion

Camera and LiDAR measurements are transformed into a common vehicle reference frame and associated using one-to-one geometric matching based on spatial separation and measurement age. Each detection can therefore participate in at most one camera–LiDAR pair during an update.

For an associated pair, an initial fused position is calculated as

$$\mathbf{p}_f = w_c\mathbf{p}_c + w_l\mathbf{p}_l, \qquad w_c + w_l = 1, \tag{F.2}$$

where $\mathbf{p}_c$ and $\mathbf{p}_l$ are the camera- and LiDAR-derived positions, respectively. The implementation gives greater weight to the LiDAR geometric measurement.

An XGBoost classifier then estimates the reliability of the fused observation using camera, LiDAR, temporal, geometric, and vehicle-state information. Two regression models estimate residual global-position corrections ΔN and ΔE, while an additional regressor estimates measurement uncertainty. The corrected fused position is

$$\mathbf{p}_{corr} = \mathbf{p}_f + \begin{bmatrix}\Delta N & \Delta E\end{bmatrix}^T. \tag{F.3}$$

Unlike the self-supervised mapping models described in Section F-B5, the XGBoost fusion models are trained using surveyed buoy positions. The corrected global position and estimated uncertainty are subsequently supplied to the AI-assisted semantic mapper.

5) AI-Assisted Dynamic Semantic Mapping

Both perception configurations use a common AI-assisted dynamic KF mapper. Each buoy landmark is represented by

$$\mathbf{x} = \begin{bmatrix}N & E & v_N & v_E\end{bmatrix}^T, \tag{F.4}$$

where N and E are the global buoy position and v_N and v_E are the estimated planar velocity components. A constant-velocity model with velocity decay propagates each landmark between observations.

Incoming measurements can only be associated with tracks having the same semantic class. Geometric consistency between measurement j and track i is evaluated using the squared Mahalanobis distance

$$d_{M,ij}^2 = \boldsymbol{\nu}_{ij}^T\mathbf{S}_{ij}^{-1}\boldsymbol{\nu}_{ij}, \tag{F.5}$$

where $\boldsymbol{\nu}_{ij}$ and $\mathbf{S}_{ij}$ are the innovation and innovation covariance, respectively.

A Random Forest classifier additionally estimates the probability p_u that the measurement represents a reliable track update using measurement- and track-state information. Candidate pairs satisfying both the geometric and learned reliability gates are ranked according to

$$J_{ij} = d_{M,ij}^2 - w_{AI}\ln\left(\max(p_{u,ij}, 10^{-3})\right), \tag{F.6}$$

followed by one-to-one detection-to-track assignment.

A second Random Forest model predicts measurement uncertainty. Its estimate σ_{AI} is blended with the incoming uncertainty σ_d according to

$$\sigma_k = (1-\alpha)\sigma_d + \alpha\sigma_{AI}, \qquad \mathbf{R}_k = \sigma_k^2\mathbf{I}. \tag{F.7}$$

The Random Forest models are trained using self-supervised labels generated from the future consistency of baseline KF tracks after a 5-s interval. For camera–LiDAR operation, the XGBoost-corrected position and uncertainty enter this second learning stage before the KF update. The camera–LiDAR configuration therefore applies learned measurement processing first during sensor fusion and again during persistent landmark association and filtering.

Unassociated observations can initialize tentative landmarks when sufficiently separated from existing same-class tracks. Repeated observations confirm landmarks, while stale tracks are removed and nearby same-class tracks can be merged to reduce duplicate map entries.

6) Semantic Map Output

Confirmed tracks form a persistent semantic map containing global position, estimated velocity, track identification, and buoy class. Each landmark can additionally be assigned an obstacle radius to provide navigation-ready information to downstream path-planning and obstacle-avoidance algorithms, including APF-based USV navigation frameworks such as [6].

C. Experimental Setup

Three experimental groups were used to evaluate the proposed framework: camera-only field mapping, physical camera–LiDAR fusion, and camera–LiDAR semantic mapping in VRX. The camera-only evaluation consisted of two field experiments. First, the camera-only mapper was tested in the field using four surveyed red and green buoys. The same observations were processed using the baseline dynamic KF and the proposed AI-assisted KF. Confirmed landmarks were matched one-to-one with surveyed buoys of the same class, and planar localization error was calculated as

$$e_i = \sqrt{(N_i - \hat{N}_i)^2 + (E_i - \hat{E}_i)^2}. \tag{F.8}$$

A second camera-only field experiment evaluated semantic-map stability with two nearby red buoys and one green buoy. Because these buoys had moved relative to their original

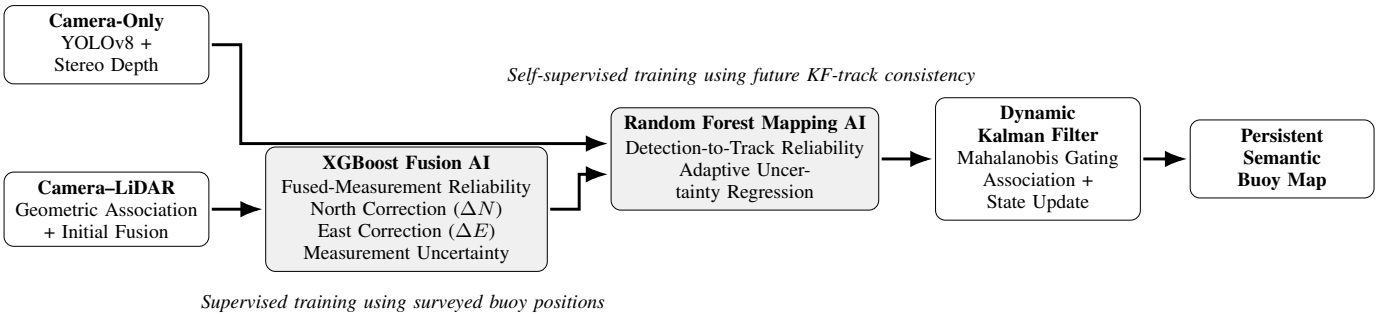


Fig. F.2. AI-assisted semantic mapping architecture combining camera-only and camera–LiDAR perception with XGBoost, Random Forest, and dynamic KF mapping.

surveyed positions, the experiment was evaluated using the correctness of the persistent semantic structure rather than absolute localization error.

Physical camera–LiDAR fusion was evaluated using surveyed red and green buoys simultaneously observed by the ZED 2i and VLP-16. A held-out subset was used to evaluate learned position correction. Because the archived field data contain individual fused measurements rather than complete KF landmark histories, these results evaluate the fusion stage rather than the final persistent-map accuracy.

Finally, the complete camera–LiDAR semantic mapping framework was evaluated in VRX [17] using 20 visible ground-truth buoys and compared with geometric camera–LiDAR fusion followed by KF mapping.

D. Results

1) Camera-Only Semantic Mapping

The self-supervised training dataset contained 20,809 detection-to-track candidate pairs, including 9,615 reliable and 11,194 unreliable observations. The Random Forest reliability classifier obtained an F1 score of 0.993 and ROC-AUC of 0.999, while the uncertainty regressor achieved a mean absolute error of 0.0185 m.

In the four-buoy field experiment, both methods maintained four matched landmarks. The baseline dynamic KF obtained a mean localization error of 1.122 m, while the proposed AI-assisted mapper reduced the error to 0.882 m. Mean frame-level RMSE decreased from 1.146 m to 0.888 m, and the maximum frame-level mean error decreased from 1.377 m to 0.991 m. These results are summarized in Table F.1.

In the close-buoy experiment, the baseline mapper intermittently generated an additional red landmark even though only two red buoys were physically present. The AI-assisted mapper maintained the expected two-red and one-green landmark structure. Figure F.3 compares the aggregated landmark positions produced by both methods. The baseline map contains three red landmarks and one green landmark, while the AI-assisted map contains the correct two red landmarks and one green landmark.

2) Camera–LiDAR Field Fusion

The supervised camera–LiDAR dataset contained 4,178 fused measurements, including 3,261 reliable and 917 unreliable observations. The XGBoost reliability classifier achieved

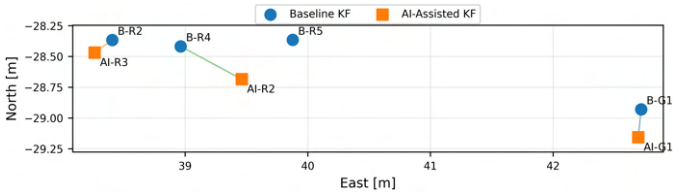


Fig. F.3. Close-buoy maps: baseline KF created an extra red landmark, while the AI-assisted KF preserved the correct structure.

an F1 score of 0.981 and ROC-AUC of 0.993. The North and East correction regressors achieved MAEs of 0.0486 m and 0.0425 m, respectively, while the uncertainty regressor achieved an MAE of 0.0203 m. The fused-data Random Forest mapper was trained separately using 3,744 detection-to-track pairs and achieved an F1 score of 0.996 and ROC-AUC of 0.999, with an uncertainty MAE of 0.0269 m.

On the held-out physical field dataset, raw fused measurements produced a mean localization error of 0.813 m and an RMSE of 0.900 m. XGBoost position correction reduced these values to 0.050 m and 0.089 m, corresponding to reductions of 93.9% and 90.1%, respectively. The results are summarized in Table F.1.

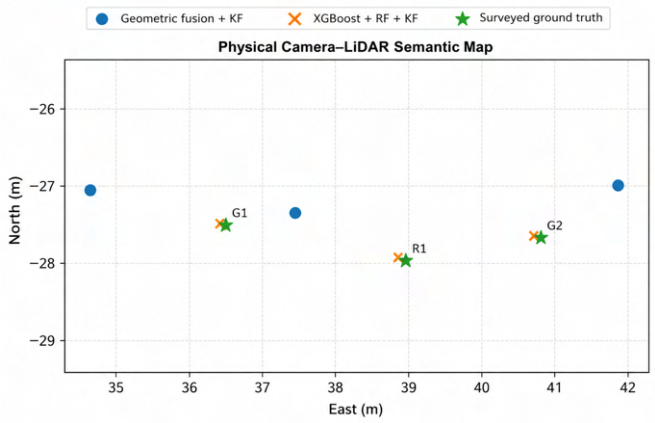


Fig. F.4. Physical camera–LiDAR semantic map comparing geometric fusion + KF, the AI-assisted framework, and surveyed ground truth.

Figure F.4 compares the geometric fusion + KF baseline and

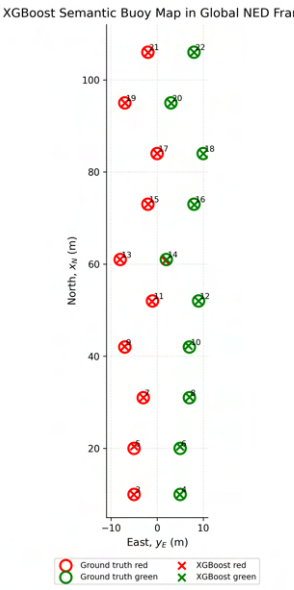


Fig. F.5. VRX semantic map for 20 ground-truth buoys, comparing mapped and true landmark positions.

the AI-assisted framework with the surveyed buoy positions. The AI-assisted estimates lie closer to the corresponding ground-truth locations for the three evaluated buoys.

3) Camera–LiDAR Semantic Mapping in VRX

The complete framework was evaluated using 20 visible ground-truth buoys in VRX. Both methods matched all 20 landmarks. Geometric fusion followed by KF mapping obtained a mean error of 0.550 m and RMSE of 0.660 m, while the AI-assisted framework reduced these values to 0.123 m and 0.180 m. Median error decreased from 0.470 m to 0.073 m and maximum error from 1.400 m to 0.459 m. The complete quantitative comparison is summarized in Table F.1.

E. Summary and Concluding Remarks

An AI-assisted semantic mapping framework combining stereo vision, LiDAR, tree-based learning, and dynamic Kalman filtering was developed for autonomous USVs. Camera-only field experiments reduced mean landmark error from 1.122 m to 0.882 m and maintained the correct two-red/one-green semantic structure in the close-buoy experiment, while the baseline intermittently created a duplicate landmark.

Physical camera–LiDAR experiments showed that XGBoost-based position correction reduced held-out fused-measurement error from 0.813 m to 0.050 m. In the 20-buoy VRX experiment, the complete framework mapped all visible landmarks while reducing mean error from 0.550 m to 0.123 m and RMSE from 0.660 m to 0.180 m.

The results demonstrate that combining learned measurement-quality estimation with model-based filtering improves both localization accuracy and semantic-map consistency. Future work will evaluate the complete framework in larger-scale closed-loop field experiments and extend the mapping architecture to additional static and dynamic maritime obstacles.

Acknowledgment

The authors acknowledge the support of the Office of Naval Research under grant N00014-23-1-2426.

During the preparation of this work, the authors used ChatGPT (OpenAI) and Grammarly solely to improve grammar, clarity, and readability. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of this publication.

USV Semantic-Mapping Paper References

[1] C. Bohm, D. Henderson, K. Mendrick, N. Murray, R. Velasco Garcia, E. I. Sarda, Z. Kassuba, R. Hildebrand, and C. Jannetta, "A USV-based platform for bathymetric mapping and underwater sound source localization," in *Proc. OCEANS Conf.*, Chicago, IL, USA, 2025.

[2] Z. Kassuba, M. Goldenbogen, C. Jannetta, X. Vicent Navarro, and E. I. Sarda, "Development of a water quality monitoring system on a USV," in *Proc. OCEANS Conf.*, Halifax, NS, Canada, 2024.

[3] A. Farrell, D. Morris, X. Vicent, and M. R. Dhanak, "Development of a sensor-enabled towfish for autonomous subsea magnetic and electric field mapping," in *Proc. OCEANS Conf.*, Monterey, CA, 2026.

[4] X. Vicent Navarro and M. R. Dhanak, "Performance and comparison of PID-based techniques for USV navigation," in *Proc. OCEANS Conf.*, Halifax, NS, Canada, 2024.

[5] X. Vicent Navarro, M. R. Dhanak, and E. I. Sarda, "PID auto-tuning for USVs: A multi-evolutionary optimization AI approach," *Ocean Engineering*, vol. 355, pt. 1, Art. no. 124987, 2026.

[6] X. Vicent Navarro and M. R. Dhanak, "Decentralized COLREGs-based navigation for multiple USVs using artificial potential fields," in *Proc. OCEANS Conf.*, Chicago, IL, USA, 2025.

[7] X. Vicent Navarro, E. I. Sarda, and M. R. Dhanak, "Fuzzy logic-assisted PID control for improved station-keeping of a catamaran-style USV," in *Proc. OCEANS Conf.*, Chicago, IL, USA, 2025.

[8] X. Vicent Navarro, C. Wilson, and E. I. Sarda, "Accurate autonomous launch and recovery of a UAV from a station-keeping USV," in *Proc. OCEANS Conf.*, Halifax, NS, Canada, 2024.

[9] L. Ma, W. Xie, and H. Huang, "Convolutional neural network based obstacle detection for unmanned surface vehicle," *Mathematical Biosciences and Engineering*, vol. 17, no. 1, pp. 845–861, 2020, doi: 10.3934/mbe.2020045.

[10] L. Trinh, S. Mercelis, and A. Anwar, "A comprehensive review of datasets and deep learning techniques for vision in unmanned surface vehicles," *Ocean Engineering*, vol. 334, Art. no. 121501, 2025, doi: 10.1016/j.oceaneng.2025.121501.

[11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.

[12] G. Welch and G. Bishop, "An introduction to the Kalman filter," Department of Computer Science, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, Tech. Rep. TR 95-041, Jul. 2006.

[13] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[14] D. Liu, J. Zhang, J. Jin, Y. Dai, and L. Li, "A new approach of obstacle fusion detection for unmanned surface vehicle using Dempster–Shafer evidence theory," *Applied Ocean Research*, vol. 119, Art. no. 103016, 2022.

[15] F. Ponzini and M. Martelli, "Camera–LiDAR data fusion for enhanced ship situational awareness in maritime environment," *Journal of Marine Science and Engineering*, vol. 14, no. 14, Art. no. 1276, 2026, doi: 10.3390/jmse14141276.

[16] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, San Francisco, CA, USA, 2016, pp. 785–794.

[17] Open Source Robotics Foundation, "VRX: Virtual RobotX," GitHub repository, [Online]. Available: https://github.com/osrf/vrx [Accessed: Jul. 1, 2026].

TABLE F.1
SUMMARY OF CAMERA-ONLY, PHYSICAL CAMERA–LIDAR, AND VRX RESULTS

Experiment	Method	Buoys	Mean (m)	RMSE (m)	Additional Metrics
Camera-only field	Baseline KF	4	1.122	1.146	Max: 1.377
	AI-assisted	4	**0.882**	**0.888**	Max: **0.991**
Physical camera–LiDAR	Raw fusion	3	0.813	0.900	–
	AI-corrected fusion	3	**0.050**	**0.089**	–
VRX semantic map	Geometric fusion + KF	20	0.550	0.660	Med.: 0.470; Max: 1.400
	AI-assisted	20	**0.123**	**0.180**	Med.: **0.073**; Max: **0.459**

[18] X. Vicent, R. Velasco, E. Diaz, C. Wilson, and M. Dhanak, "Navigating dynamic waters: Learning-based control and potential-field planning for ASVs," Team Owltonomous, RoboBoat 2026 Technical Design Report, 2026.
[19] Ocean Power Technologies, "WAM-V 16 Unmanned Surface Vehicle," [Online]. Available: https://oceanpowertechnologies.com/products/unmanned-surface-vehicles/wam-v-16/ [Accessed: Jul. 1, 2026].
[20] Stereolabs, "How to Use YOLO with ZED," *ZED SDK Documentation*. [Online]. Available: https://docs.stereolabs.com/docs/integrations/yolo [Accessed: Aug. 17, 2026].
[21] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, Shanghai, China, 2011, pp. 1–4, doi: 10.1109/ICRA.2011.5980567.

APPENDIX G
UAV TESTING PLAN

A. Scope

The UAV Testing strategy started with simulation testing offline before physical testing on hardware. The RoboNation Data Sharing resources were used to acquire base models in Gazebo which were converted to formats that could be used in IsaacSIM. This was used to train preliminary models and algorithms for use on hardware. Early development used hardware offboard the drone before testing models and algorithms in flight.

B. Schedule

The testing schedule is organized around four primary phases: simulation-based development, offline validation using recorded datasets, hardware validation using off-board systems, and in-flight system integration. Table G.1 summarizes the testing timeline, objectives, and environments.

Integrated Mission Testing Via IsaacSIM

C. Resource & Tools

Testing activities leverage a combination of software and physical resources. Simulation and offline testing are conducted using the IsaacSIM 4.5 on dedicated virtual workstation. Perception development relies on datasets from the RoboNation Data Sharing platform modified to work in IsaacSIM. Hardware development and integration are supported by FAU's CAAI laboratory.

For on-water testing, the UAV is deployed over a pool and beacons are controlled using an Arduino nano with signal sent via a 25 ft. wire. Data collected during physical training was used to finetune models and improve training.

D. Environment

Software testing is initially conducted in simulation and using shared datasets due to limited access to competition-grade physical objects. This approach enables validation of perception algorithms and mission logic prior to physical deployment.

On-water testing is conducted at FAU or at team member's personal property. Airspace at FAU's SeaTech restricts fligth without extensive approvals, requiring an alternative for regular flight testing. Local regulations are verified prior to each flight test to ensure the UAV does not interfere with external traffic. Weather conditions are also verified to ensure safe flight.

E. Risk Management

Safety considerations are integrated into all testing activities. Battery handling and charging procedures follow established protocols to mitigate risks associated with lithium-polymer batteries. All work conducted in the electronics and machine shops adheres strictly to FAU laboratory safety guidelines.

Local flight restrictions are checked through Federal Aviation Administration (FAA) notices to airmen (NOTAM) to make sure there are no temporary flight restrictions prior to flight. Safe weather conditions are also verified using weather reports in 1800wxbrief. Water testing was done in controlled environments to ensure safety of the environment and to mitigate risks to team members. The UAV was equipped with flotation safety gear when operating near water for testing. All members of the UAV team were briefed with safety procedures prior to flight.

APPENDIX H
BEACON DETECTION ALGORITHM AND TESTING ON UAV

Autonomous Aerial Scouting in Maritime Environments for RobotX Search and Rescue Mission

Samantha Garcia
Center for Connected Autonomy & Artificial Intelligence
College of Electrical Engineering & Computer Science
Florida Atlantic University
Boca Raton, FL, USA
scruz10@fau.edu

TABLE G.1
TEAM SEAOWLS UAV TESTING SCHEDULE AND RESULTS

Start/End Date	Documentation	Target	Result	Environment	Member Presence
07/01/25 - 08-31-25		YOLO vision model training using simulated images	$\geq$ 0.71 confidence in competition objects	SeaBird Machine, IsaacSIM	Data Science Team
09/15/25 - 10/30/25		Lawnmower pattern and reliable detection	Reliable detection of buoys and autonomous flight in simulation	SeaBird Machine, IsaacSIM	Data & Computer Science Teams
11/10/25 - 12/30/25		Training and testing on physical beacons images	Retraining required from simulation model	Jetson Orin Nano & SeaBird machine	Data Science, Software & Controls Team
01/10/26 - 02/10/26		In flight testing	Custom drone unable to fly, Distance Estimation unreliable in COTS drone	Simulation	Data Science, Software & Controls Team
02/20/26 – 03/31/26		Integrated mission testing;	End-to-end autonomy validation	ModalAI Starling 2 Max Drone	Full Team
04/10/26-05/20/26		On-water system testing with USV	Full system integration under disturbances	On-Water tests with ModalAI Starling 2 Max Drone	Full Team;

Adriel Marie Abad
Center for Connected Autonomy & Artificial Intelligence
College of Electrical Engineering & Computer Science
Florida Atlantic University
Boca Raton, FL, USA
aabad2022@fau.edu

Bob Varghese
Center for Connected Autonomy & Artificial Intelligence
College of Electrical Engineering & Computer Science
Florida Atlantic University
Boca Raton, FL, USA
bvarghese2026@fau.edu

A UAV framework that autonomously searches from random takeoff locations, detects flashing maritime beacons, classifies their color and blink state, geolocates them using GPS, and transmits results to a ground station.

Abstract— Autonomous, multi-domain robotic systems have the potential to make complex maritime operations such as search-and-rescue safer and more efficient. However, current vision frameworks struggle in dynamic maritime environments which often present challenging lighting conditions like water reflection and refraction. Moreover, they fail to detect colored temporal signals that maritime beacons often rely on to convey water hazards and environmental conditions. We propose a novel, model-agnostic vision framework for identifying color and flashing frequency using a fast, downstream script, avoiding expensive model retraining and architectural changes. Moreover, the drone's real-world performance is tested through waypoint missions. These methods are evaluated with preliminary real-world testing deployed physically on a drone, with the vision models benchmarked against stationary scenes and fast-moving image captures. Our results demonstrated a promising foundation, with over 80% color accuracy and 100% blinking detection precision on waypoint missions. These results prove that our framework provides a strong basis for future works undertaking maritime autonomous UAV operations.

Index Terms—UAVs, autonomous mission, autonomous navigation, object detection, exploration.

A. Introduction

Fully autonomous robotic systems have garnered attention within military and civil applications in the interest of increasing operational efficiency, mitigating human error, and protecting material assets and human lives. However, complex maritime operations such as search-and-rescue, tactical reconnaissance, and environmental monitoring demand multi-domain robotic collaboration.

Robots have inherently varied capabilities that enable them to excel in isolated tasks. Unmanned surface vessels (USVs) have better endurance and higher payload capacity, yet strug-

gle to navigate with a flat line-of-sight that is easily obstructed by waves and obstacles. Meanwhile, unmanned aerial vehicles (UAVs) are superior in scouting, yet cannot in themselves perform the search and rescue as their agility lies in their lightweight construction. A successful mission relies on the robots' smooth and successful coordination and communication. In a search-and-rescue scenario, such as one simulated by RobotX's "Safe Passage" task, the USV relies on the UAV's perception capabilities to identify and signal dangers. The UAV plays a crucial role in ensuring the USV's safe navigation through hazardous waters, and inaccurate or incomplete information can be costly.

Maritime environments are unique in their challenges. Maritime beacons depend on visual signals to convey a wide, diverse variety of hazards and water conditions. Current vision frameworks struggle to identify and differentiate among these signals due to challenging lighting conditions like water reflection and refraction which obscure vision, as well as temporal signals like flashing frequencies that cannot be captured by vanilla object detection models as they analyze frame-by-frame.

This paper proposes a novel framework capable of navigating hazardous maritime environments within UAV-USV cooperative operations to undertake complex, autonomous search-and-rescue missions. Specifically, we make the following contributions:

1) A lightweight two-stage model-agnostic perception pipeline capable of beacon color and light blinking detection, performed without retraining vision models with new architectural backbones;
2) Proof-of-concept validation of our methods through real-world testing and deployment on a physical drone following waypoint missions.

This paper investigates related work to identify gaps within the literature we aim to bridge, describes our approach in detail, presents the experimental results obtained through real-world testing, and discusses the performance, limitations and future work.

B. Related Work

This section aims to review current research in these three core areas: spatiotemporal computer vision, autonomous area exploration in unknown environments, and the simulation-to-reality gap between simulation and physical deployment.

1) State of the Art

Some works have addressed existing computer vision models' weakness in temporal processing. [1] moved away from performing object detection on static images, proposing novel work that provides spatiotemporal memory for video object detection. [2] also proposes a novel design that uses a foreground detector to compute a per-pixel probability map that informs a temporal dimension to the neural network, but is critically limited to a static camera. Though not using object detection, adjacent tracking frameworks have also explored temporal variations—for instance, real-time blink detection software that were implemented to measure physical light fluctuations or periodic human optical changes [3].

For autonomous navigation within unknown environments, modern UAV exploration leverages hybrid frontier-guided and greedy search algorithms to optimize path planning and maximize mapping coverage [4]. An existing product that directly addresses this with a UAV is Exyn's product, a GPS-denied drone capable of mapping GPS-denied areas with Simultaneous Localization and Mapping (SLAM) [5]. Though it does not provide a comprehensive deep-dive into the AI architecture, it is mentioned to be specifically effective for search-and-rescue missions in vast, unpredictable environments.

As these drones typically operate in complex environments and involve heavy software stacks, researchers rely heavily on high-fidelity simulators to reduce the risk of field-testing. [6] makes the case for using simulations to learn policies for exploration in the real-world with reinforcement learning when direct sim-to-real transfers are not viable. Following this principle, [7] uses deep reinforcement learning for VTOL-UAVs that can learn a generalized policy that is effective in both simulation and real settings.

2) Addressing Literature Gaps

Existing literature does not provide a synthesis of these disparate research domains into a single, cohesive aerial reconnaissance framework. Prior literature has addressed these critical areas separately, but this project successfully bridges them. For example, Exyn's GPS-denied drone performs autonomous search with state of the art sensors like LiDAR and offers a 360-degree visual field of view, but its proprietary AI suffers from the same weaknesses of traditional vision models in that they are not capable of processing temporal information [5]. This is especially valuable in maritime environments where colored buoys with varied light rhythms are commonly used to signal conditions in the water, such as hazards and safe routes [8].

Moreover, rather than modifying or retraining heavy deep-learning architectures to handle a temporal dimension, we introduce a highly efficient, post-processing spatiotemporal filter. Our pipeline uses a standard object detection algorithm like YOLO to recognize different colored beacons while delegating the task of differentiating solid and blinking lights to a lightweight downstream script. This is far more computationally efficient and is compatible for any standard object detection model.

Towards bridging the sim-to-real transfer, we use a powerful 3D simulator, IsaacSIM, which can model environments and objects in realistic detail. Simulating beacons that closely resemble their real-world counterparts enables the rapid generation of massive amounts of diverse images to train the object detection models. Moreover, a physical reconstruction of the maritime beacon enables real-world data collection for further fine-tuning, thus closing the sim-to-real gap.

The most closely related work is [9]'s autonomous drone for search and rescue in forests. Rather than operating in the challenging conditions of a maritime environment, it instead tackles dense, heavily-occluded forests. It is also meant to

perform a search and rescue mission, similar to the RobotX mission in which our project is centered. However, our work stands out in its necessity and ability to process temporal visual signals and tackles difficult problems related to the waters — water reflection and unpredictable wave patterns.

C. Methodology

The beacon detection algorithm consisted of three distinct tasks:

1) Beacon detection
2) Beacon classification
3) Geolocation of beacon

Beacon detection and classification were initially combined into one task, however it became necessary to separate them to accurately determine if a beacon was blinking or steady. By detecting beacons in general, rather than colored beacons, an image processing pipeline could then be used to determine the beacon color as well as if the beacon was blinking.

1) Beacon detection

The first task was beacon detection. A YOLOv8n model was initially trained in a simulated environment through IsaacSIM (version 5.1.0), modeling a simple maritime port. The maritime beacons are designed to closely resemble the real-life prototype to ensure proper sim-to-real transfer. Robust domain randomization was performed to ensure the model was trained across diverse scenes to identify beacons; beacon locations, camera angle, camera altitude, camera location, time of day, and water appearance were randomized in each image with the potential of some or no beacons to be captured. 3000 diverse images were generated and used for training, split 60/20/20 across training, testing, and validation.

These models were then fine-tuned with a custom auto-labeling software using video footage of a beacon taken from a *DJI mini 3* drone and photos taken from the *ModalAI Starling2* drone. After the model accurately detected the beacon, the frame was cropped around the beacon. Images of the beacon were then used to train a second model to detect the beacon tops.

The beacon top YOLOv8 model was trained using only images of beacons using the custom auto-labeling software. A self-supervised learning system was used by manually labeling a small set of images to create an initial model for labeling the larger set of images. Figure H.1 shows a beacon and beacon top detection, with the beacon top shaded in blue.

2) Beacon classification

Beacons needed to be classified both by color and blinking status. This classification used an image of a cropped beacon top from the detection algorithm, which was then converted to hue-saturation-value (HSV) format. The hue was used to determine the beacon color, while saturation and value were used in combination to determine the intensity of light. The beacon colors were classified as red, green, blue, white, or unknown. White and unknown colors were used when the beacon light was considered "off," however the beacon often appeared to be blue when it was actually off. This made the intensity of light necessary to determine if the beacon was on or off.

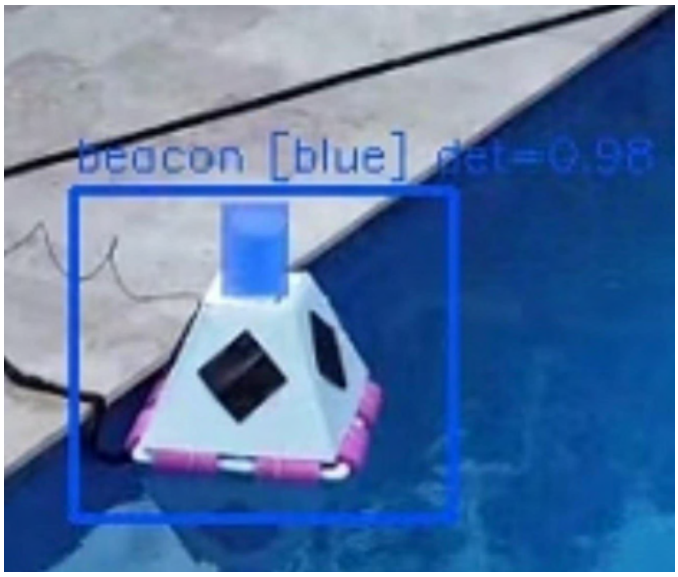


Fig. H.1. Detected beacon and classified beacon top LED color.

Blinking status was determined by the frequency with which the light changes. For red and green beacons, the change in color could be used to determine if the beacon was blinking. For blue beacons, the light intensity was used as a fallback to determine the blinking status since the color may not appear to change in certain environments.

a) Intensity and Color Calculation

Saturation and value minimums were used to determine which pixels were part of the light by creating a light mask where any pixels above the minimum saturation and value are set to 1, and all other pixels are 0, as shown in (H.3).

$$S_{min} = 40 \tag{H.1}$$

$$V_{min} = 80 \tag{H.2}$$

$$\text{light_mask} = 255 \cdot \mathbf{1}\left[s \geq S_{\min} \wedge v \geq V_{\min}\right] \tag{H.3}$$

Then, the number of lit pixels were calculated by finding the total bits in the light mask. The color confidence was calculated as the number of lit pixels divided by the total number of pixels. This is shown in (H.4) and (H.5).

$$N_{\text{lit}} = \sum \mathbf{1}\left[\text{light_mask} > 0\right] \tag{H.4}$$

$$cc = \frac{N_{\text{lit}}}{\max(N_{\text{total}}, 1)}, \qquad N_{\text{total}} = W \times H \tag{H.5}$$

Equations (H.6)-(H.8) show how pixels with values higher than 220 were classified as very bright, and therefore consid-

ered white pixels.

$$\text{very_bright}_i = \mathbf{1}\left[v_i \geq 220\right] \tag{H.6}$$

$$N_{\text{bright}} = \sum_i \text{very_bright}_i \tag{H.7}$$

$$\text{intensity}_{\text{white}} = \frac{1}{255\, N_{\text{bright}}} \sum_{i\,:\,\text{very_bright}_i = 1} v_i \tag{H.8}$$

The intensity for non-white pixels calculated in (H.9) was measured by summing the value of any pixels in the light mask, dividing by 255, and dividing by the total number of lit pixels. This gave a scale between 0 and 1 for intensity value.

$$\text{intensity} = \frac{1}{255} \cdot \frac{1}{N_{\text{lit}}} \sum_{i\,:\,\text{light_mask}_i > 0} v_i \tag{H.9}$$

Hue statistics were calculated as shown in (H.10)-(H.13). These were used to determine color classification as well as debugging for misclassified colors and blinking status related to color classification.

$$\bar{h}_{\text{mean}} = \frac{1}{N_{\text{lit}}} \sum_{i=1}^{N_{\text{lit}}} h_i \tag{H.10}$$

$$\sigma_h^2 = \frac{1}{N_{\text{lit}}} \sum_{i=1}^{N_{\text{lit}}} \left(h_i - \bar{h}_{\text{mean}}\right)^2 \tag{H.11}$$

$$h_{\text{median}} = \text{median}\left(\{h_i\}_{i=1}^{N_{\text{lit}}}\right) \tag{H.12}$$

$$h_{\text{mode}} = \underset{k \,\in\, \text{unique}(\{h_i\})}{\arg\max} \; |\{\, i : h_i = k \,\}| \tag{H.13}$$

The hue mode was used to select color based on the default bands given in (H.14). Slight modifications were made to the hue band values based on the camera being used. It became apparent that different camera quality had an effect on how colors were classified, and a configuration file was used to modify hue bands, as well as other system specific features.

$$\text{band}(h) = \begin{cases} \text{red} & \text{if } h \in [0^\circ, 20^\circ] \\ \text{green} & \text{if } h \in [35^\circ, 95^\circ] \\ \text{blue} & \text{if } h \in [105^\circ, 134^\circ] \\ \text{red} & \text{if } h \in [170^\circ, 180^\circ] \end{cases} \tag{H.14}$$

b) Blink Detection Status

A window of at least six samples was used to determine if a beacon was blinking or not. Since beacons that are off are frequently detected as blue beacons, these cases needed to be treated differently when determining blinking status. Before blinking status was assigned, (H.15) shows how the color of the beacon in the window was determined to be blue if at least half of the beacons in the window were blue or unknown color.

$$\text{is_blue}(W) = \mathbf{1}\left[\sum_{i \,\in\, W} \mathbf{1}\left[c_i \in \{\text{blue}, \text{unknown}\}\right] \;>\; \frac{|W|}{2}\right] \tag{H.15}$$

A gap in detections was also used to determine if a beacon was in the "off" status. If there was more than 5 seconds (H.16) between detections, an "off" beacon status was inserted at the midpoint of the window (H.17).

$$T_{\text{gap}} = 5 \tag{H.16}$$

$$\text{if } \neg\,\text{is_blue} \;\wedge\; (t_{\text{new}} - t_{\text{last}}) > T_{\text{gap}}, \quad \text{insert } \left(t_{\text{last}} + \tfrac{T_{\text{gap}}}{2},\ \text{off}\right) \tag{H.17}$$

If the gap exceeded more than 12 seconds, the off blink status was discarded. Detections were paused until another beacon top was detected to start a new window. The time between detections was used to validate blinking status. The time between the first and last detection in the window needed to be at least 4 seconds, and there needed to be at least 4 detections before blinking status could be determined.

The beacon's color in the window was used to determine whether to use color change frequency or intensity frequency for blinking determination. If the beacon was red or green, color changes were used, while blue beacons used light intensity. If the beacon was determined to be red or green, a red or green detection was considered an "on" phase, while blue or unknown was "off".

To determine if a blue beacon was on, the color confidence calculated in (H.5) was used as well as the the mean intensity (H.20) across the window. When the color confidence was above 0.15, a blue beacon was considered "on." In addition, if the intensity swings above the mean intensity, the beacon was considered "on" (H.21). Below the mean returned an "off" phase. For intensity calculation, a minimum swing (H.19) between the highest and lowest intensity needed to be at least 0.05.

$$L = \{\, i : c_i \notin \{\text{off unknown}\}\,\} \tag{H.18}$$

$$\text{swing} = \max_{i \in L}(I_i) - \min_{i \in L}(I_i) \tag{H.19}$$

$$\bar{I} = \frac{1}{|L|} \sum_{i \,\in\, L} I_i \tag{H.20}$$

$$\text{on}_i = \mathbf{1}\left[i \in L\right] \cdot \mathbf{1}\left[I_i \geq \bar{I}\right] \tag{H.21}$$

$$\text{phase} = \begin{cases} \text{on} & \text{if } \text{on}_N = 1 \\ \text{off} & \text{if } \text{on}_N = 0 \end{cases} \tag{H.22}$$

To ensure that blinking was consistent, a minimum number of rising edges for each frequency was set to 3 for blue beacons and 2 for other colors (H.23), (H.25). A minimum edge gap of 0.2 was used to remove noise that may give inaccurate blinking status (H.24).

$$E_{\text{raw}} = \{\, t_i \;:\; \text{on}_{i-1} = 0 \;\wedge\; \text{on}_i = 1,\;\; i = 2, \ldots, N \,\} \tag{H.23}$$

$$E = \{\, t \in E_{\text{raw}} \;:\; t - t_{\text{last edge}} \geq \Delta t_{\min} \,\}, \quad \Delta t_{\min} = 0.2 \tag{H.24}$$

$$n_{\min} = \begin{cases} 3 & \text{if blink_color} = \text{blue} \\ 2 & \text{otherwise} \end{cases}, \quad \text{insufficient if } |E| < n_{\min} \tag{H.25}$$

As an additional fail-safe, non-blue beacons were detected as steady if their "on" to "off" ratio was greater than 0.8 (H.26). The inter-onset interval (IOI) was calculated by finding the gap between consecutive edges (H.27). Then, the mean IOI was calculated to ensure that it was greater than 0 (H.28).

$$\text{on_fraction} = \frac{1}{N}\sum_{i=1}^{N} \text{on}_i, \quad \text{reject if } \text{on_fraction} > 0.80 \tag{H.26}$$

$$\text{IOI}_j = e_{j+1} - e_j, \qquad j = 1, \ldots, |E| - 1 \tag{H.27}$$

$$\overline{\text{IOI}} = \frac{1}{|E| - 1}\sum_{j=1}^{|E|-1} \text{IOI}_j \tag{H.28}$$

$$\text{reject if } \overline{\text{IOI}} \leq 0 \tag{H.29}$$

A maximum IOI was set to 5 seconds for blue beacons and 8 seconds for colored beacons. Any IOI greater than the maximum was rejected (H.30).

$$\text{IOI}_{\max} = \begin{cases} 5 & \text{if blue} \\ 8 & \text{otherwise} \end{cases}, \quad \text{reject if } \max_j (\text{IOI}_j) > \text{IOI}_{\max} \tag{H.30}$$

Once it was determined that a window of detections was blinking, the inverse of the mean IOI was used to calculate the blinking frequency (H.31). While the frequency expected was 1 Hz, the minimum and maximum frequency was set to between 0.12 Hz and 2.0 Hz (H.32) since latency from processing steps sometimes affected frequency measurements.

$$f_{\text{blink}} = \frac{1}{\overline{\text{IOI}}} \tag{H.31}$$

$$\text{is_blinking} = \mathbf{1}\left[f_{\min} \leq f_{\text{blink}} \leq f_{\max}\right], \quad (f_{\min}, f_{\max}) = (0.12, 2.0) \tag{H.32}$$

3) Geolocation

Geolocating beacons required determining the distance the beacon was from the UAV. Once the distance was measured, the GPS location of the beacon was retrieved from a ROS topic as well as the UAVs orientation. The ideal camera choice was a ZED stereo camera, however the modalAI starling2 drone processor did not meet minimum requirements for the ZED SDK. Instead, the onboard monocular camera was used for detection, and therefore a different strategy was required in order to estimate the beacon's distance.



(a) ArUco marker beacon setup



(b) ArUco and beacon image from the UAV

Fig. H.2. Beacon with ArUco marker for distance estimation: (a) beacon setup and (b) aerial image.

a) Distance Measurement

ArUco markers [10] were used to get a ground-truth measurement for the distance between the UAV and the beacon, as shown in Fig. H.2(a). While this is a well-established method for distance estimation on monocular cameras, the distance from the beacon required a larger aruco marker in order to be detected and measured.

Despite the difficulty of using the arUco markers, a similar method was able to be implemented to estimate the distance of the beacon using the real-world size and the bounding box of the beacon. The same configuration file used for the arUco marker was used, and the camera settings were included in the beacon detector's configuration file. While this method has been introduced into the system, it is yet to be tested in the field.

b) Triangulation

Once the distance estimate is calculated, this was then used for geolocating objects by triangulation. The height of the UAV was retrieved from a ROS topic as well as its GPS location. The drone's pose is calculated using the quaternion matrix for the drone, the distance to the beacon, and the beacon's position on the camera frame, as shown in (H.33).

$$\mathbf{p}_{\text{world}} = R_{\text{body}\to\text{world}}\left(R_{\text{body}\to\text{cam}}^{\top}\,\mathbf{p}_{\text{cam}} + \mathbf{o}_{\text{mount}}\right) + \mathbf{p}_{\text{drone}} \tag{H.33}$$

4) Experiment Setup

The beacon detection algorithm was tested with two different setups. The first was with a Jetson Orin Nano and a stationary ZED stereo camera. The second set of testing was done on the ModalAI starling2 while flying a waypoint mission.

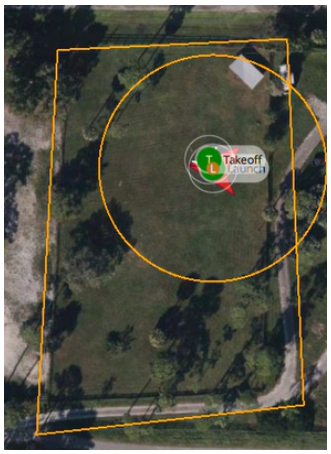
a) Experimental Protocol

The stationary evaluation included six trials, with one trial performed for each beacon configuration. The airborne evaluation included seven trials: one trial for five configurations and two trials for the red-blinking configuration. In total, thirteen trials were conducted across both experimental setups.

Each trial produced multiple detections over time. These detections were used to calculate frame-level confusion matrices and classification metrics. Detections collected during the same trial shared the same beacon configuration, camera



(a) Jetson Orin and ZED camera setup



(b) ModalAI waypoint mission

Fig. H.3. Experimental configurations used for beacon-algorithm testing: (a) stationary-camera configuration and (b) UAV waypoint mission.

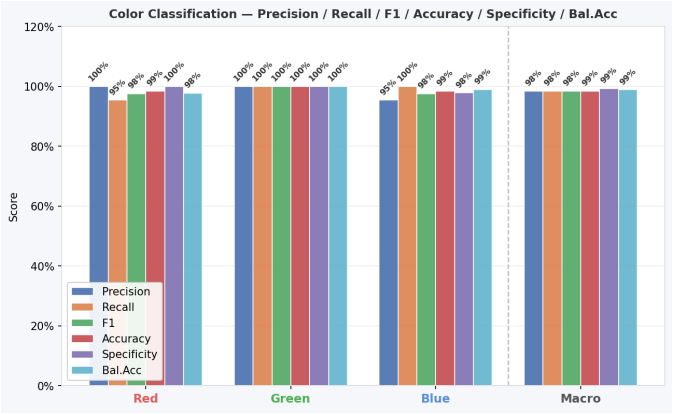


Fig. H.4. Color classification metrics using stationary ZED camera with Jetson Orin Nano.

trajectory, illumination, background, and system state. They were therefore temporally correlated and were not treated as independent experimental replicates.

Because only one trial was completed for most configurations, the reported metrics are descriptive measurements of system performance during the observed trials. They should not be interpreted as estimates of performance across repeated missions or a broader range of environmental conditions.

b) Stationary Camera Testing

The stationary tests were done with the Jetson Orin Nano and ZED cameras facing the beacon at about 15 meters away as displayed in Fig. H.3(a). The weather was sunny with clear skies. A docker container was used to start ROS and include all ZED SDK requirements. Inside the container, six shell scripts were used to run the beacon detector and log ground truth for each run. For each run, the beacon was set to the specified ground truth for the entire test. Each beacon configuration was tested in one independent run, producing six total runs. The algorithm ran until there were at least 50 beacon detections, returning a total of 300 detections across all trials. During each test, the time of detection, beacon color, blinking status, size, color intensity, percentage of each color detected, detection confidence, estimated distance and location, tracking id, blink frequency, blink phase, target color, and target blink status were logged into a CSV file for later analysis.

c) ModalAI Waypoint Mission Testing

The image in Fig. H.3(b) shows a simple waypoint mission where the ModalAI Starling2 drone flew about 1 foot in front of the takeoff location before hovering at 10 ft altitude for 30 seconds and returning to the takeoff location. The weather was sunny with clear skies. Geofence areas were set up around a field to prevent the drone from flying to unexpected areas. Failsafes in place told the drone to return to the takeoff location if the boundaries were violated. The same waypoint mission was used for all tests with the drone. There were two trials completed with the red blinking beacon, and one trial for each other beacon configuration, producing a total of seven trials with two hundred seventy-five total detections. To ensure the pipeline runs properly, several system services were developed in order to run the drone's native voxl framework with ROS2 and the custom software developed. The first service activated the voxl-mpa-to-ros2 node to allow the script to subscribe to ROS camera topics through a Data Distribution Service (DDS). The second started a docker container using the same DDS profile as the voxl-mpa-to-ros2 service and started the beacon detector script. Like in the stationary tests, six scripts were used for each ground truth, and the beacon was set to the expected configuration before each waypoint mission. Each trial was run for about 45 seconds total.

D. Results

The data used for this report were collected from live testing. Images of the beacon and the beacon top were captured and logged as objects were detected. Data was collected and logged in a CSV as described in H-C4b for all test. Data from testing is publicly available as referenced in the data availability section.

The data collected from testing were evaluated against ground truth for each beacon configuration. The precision, recall, accuracy, F1 score, specificity, and balanced accuracy were calculated to measure how well the system detected color and blinking status. To give more perspective on results, the latency between detections was also measured.

a) Stationary Camera Results

The stationary camera results were promising for both the color classification and blinking status detection. Figure H.4 shows the precision, recall, f1, accuracy, specificity, and balanced accuracy for each color, as well as the macro scores. The color accuracy was 99% for red, 100% for green, and 98% for blue. The balanced accuracies were also very promising, with 98% for red; 100% for green; and 99% for blue. Precision, recall, and F1 score, and specificity were 95% or higher for all colors, showing there were very few incorrectly identified colors. The overall balanced accuracy was 99% across all colors.

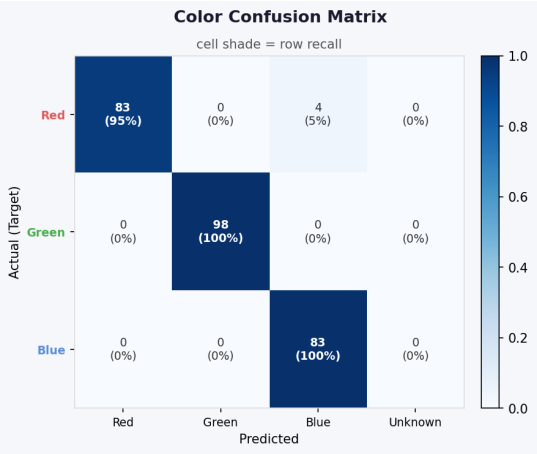


Fig. H.5. Color classification confusion Matrix using stationary ZED camera with Jetson Orin Nano.

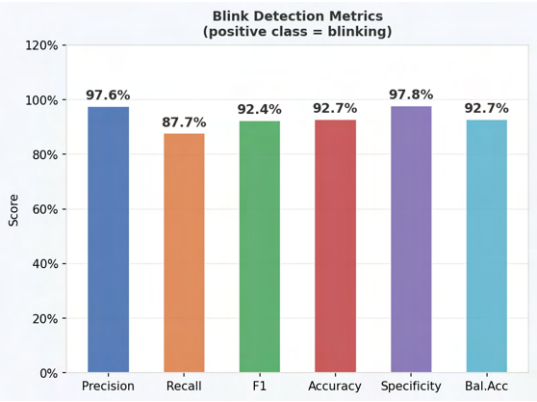


Fig. H.6. Blink detection metrics using stationary ZED camera with Jetson Orin Nano.

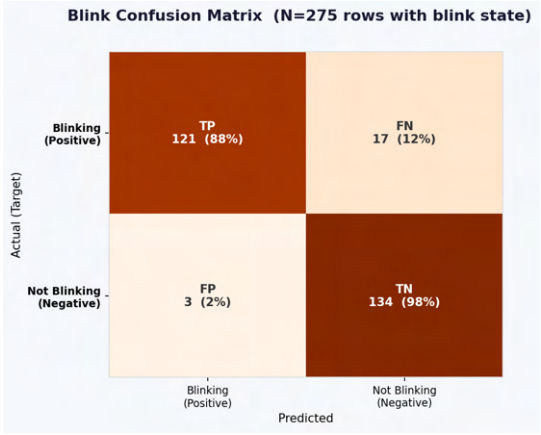


Fig. H.7. Blink detection confusion matrix using stationary ZED camera with Jetson Orin Nano.

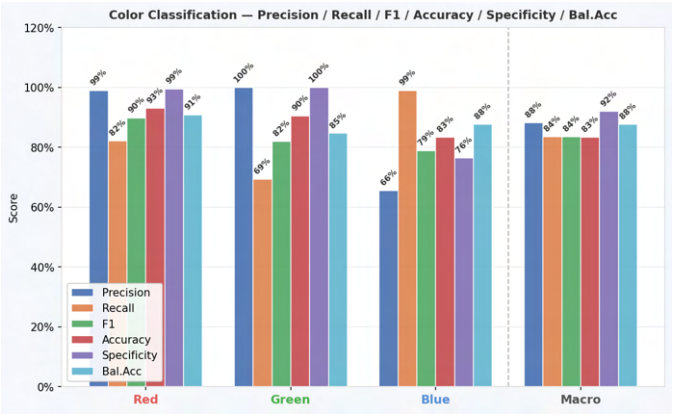


Fig. H.8. Color classification metrics using waypoint mission on ModalAI Starling 2 Max.

The confusion matrix shown in Figure H.5 gives a breakdown of how the colors were detected. The only discrepancy was the 4 beacons that were red, but classified as blue.

The blink detection was also reliable with the stationary camera setup. The accuracy was 92.7% and balanced accuracy was 92.7%, as displayed in Figure H.6. The precision, recall, and F1 scores, and specificity were 97.6%, 87.7%, 92.4%, and 97.8%, respectively. This shows that there were more true negatives than true positives.

The confusion matrix shown in Figure H.7 supports the results shown in Figure H.6. There were 121 true positives and 134 true negatives. Conversely, there were only 3 false positives and 17 false negatives. The beacon detector had more difficulty detecting when the beacon was blinking than when it was not, but overall results were positive.

The summary in Table H.1 gives an overview of metrics for each color as well as blinking status.

The average latency between detections when using the ZED cameras was 1505 ms, with a standard deviation of 237 ms. Given that the latency is higher than the blinking rate, the detection accuracy is surprising. The Nyquist rate for this would be 500 ms for a light blinking at 1 Hz. This shows the need for a more efficient pipeline and more validity testing.

b) ModalAI Waypoint Mission Results

The results from the flight testing were less promising than the stationary testing. The color detection was somewhat reliable, however the blink detection was highly inaccurate.

The bar chart in Figure H.8 shows the precision, recall, f1, accuracy, specificity, and balanced accuracy for each color detected. While the precision was 99% for red and 100% for green, the precision for blue was only 66%. This is likely due to instances when the beacon was red or green in an off phase. This is supported by the low green recall at 69%. Despite these lower values, the balanced accuracies are 91% for red, 85% for green, and 88% for blue. Overall, the macro balanced accuracy was 88%.

The confusion matrix for color in Figure H.9 gives a better idea about why the blue classification was unreliable. There were 31% of the green beacons that were classified as blue, and only 18% of red beacons. Interestingly, there was one blue beacon that was classified as red, but this is likely just an anomaly.

The blink detection metrics shown in Figure H.10 were

TABLE H.1
COLOR AND BLINK DETECTION RESULTS SUMMARY FOR STATIONARY ZED CAMERA WITH JETSON ORIN NANO

	Precision	Recall	F1	Accuracy	Specificity	Balanced Accuracy
Red	100.0%	95.4%	97.6%	98.5%	100%	98%
Green	100.0%	100.0%	100.0%	100.0%	100%	100%
Blue	95.4%	100.0%	97.6%	98.5%	98%	99%
Macro avg	**98.5%**	**98.5%**	**98.4%**	**98.5%**	**99%**	**99%**
Blink	97.6%	87.7%	92.4%	92.7%	97.8%	92.7%

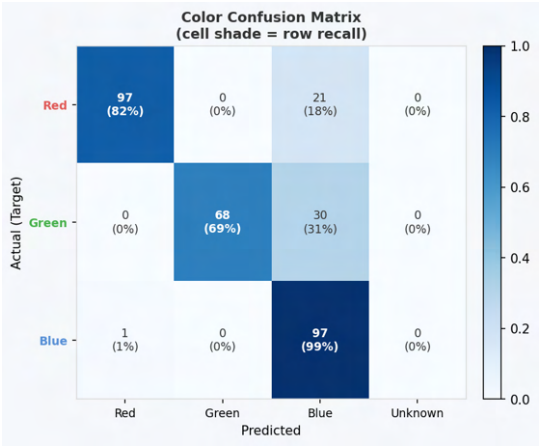


Fig. H.9. Color classification confusion matrix using waypoint mission on ModalAI Starling 2 Max.

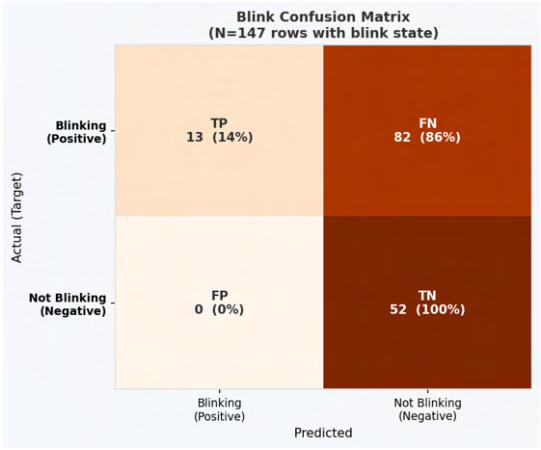


Fig. H.11. Blink detection confusion matrix using waypoint mission on ModalAI Starling 2 Max.

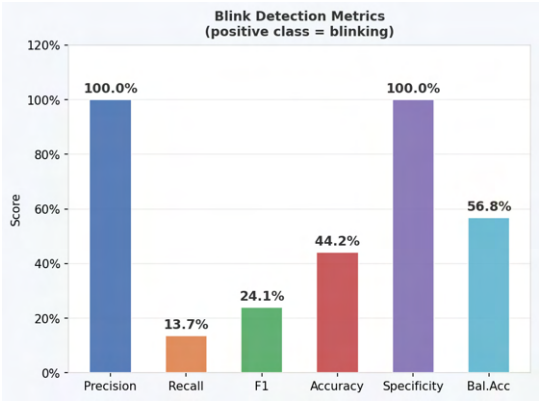


Fig. H.10. Blink detection metrics using waypoint mission on ModalAI Starling 2 Max.

highly unreliable during flight testing. While the specificity was 100%, there was only 13.7% recall and 44.2% accuracy. This resulted in a balanced accuracy of only 56.8%. The F1 score was only 24.1%. This indicates that the detector almost always showed that the beacon was not blinking, despite half of the testing being done with a blinking beacon.

The confusion matrix in Figure H.11 shows the predicted blinking status compared to ground truth. The 86% false negatives is expected based on the accuracy and recall shown in Figure H.10.

Table H.2 shows an overview of the metrics from the flight testing, showing that there is significant work to be done to improve blink detection status.

The average latency for the flight tests was much higher at 2147 ms. More surprisingly, the standard deviation was 3533 ms, showing that the time between detections was extremely inconsistent, possibly contributing to the blink detection problems.

E. Discussion

The two experimental configurations described in Section H-C produced markedly different outcomes, and the gap between them is the most informative result of this work.

1) Color Classification and the Off-Phase Ambiguity

Color detection operates per-frame, therefore its errors are bounded. The confusion matrices in Figures H.5 and H.9 show that all misclassifications are directional, with red and green beacons misidentified as blue during their off phases due to the translucent housing reflecting ambient sky lighting.

The pipeline avoids the issues of per-frame color detection by carrying color forward through the temporal window described in (H.15). The limitation is that this correction depends on the temporal stage functioning, which on the airborne platform it largely did not.

2) Sampling Rate, Aliasing, and the Blink Collapse

The blink results initially appeared contradictory. The stationary configuration achieved 92.7% blink detection despite a 1505 ms inter-detection latency, well above the 500 ms Nyquist

TABLE H.2
COLOR AND BLINK DETECTION RESULTS SUMMARY USING WAYPOINT MISSION ON MODALAI DRONE

	Precision	Recall	F1	Accuracy	Specificity	Balanced Accuracy
Red	99.0%	82.2%	89.8%	93.0%	99%	91%
Green	100.0%	69.4%	81.9%	90.4%	100%	85%
Blue	65.2%	99.0%	78.9%	83.4%	76%	88%
Macro avg	**88.2%**	**83.5%**	**83.5%**	**83.4%**	**92%**	**88%**
Blink	100.0%	13.7%	24.1%	44.2%	100%	56.8%

threshold for a 1 Hz signal. However, the low latency variance enabled edge-counting logic to detect state changes consistently. Highly irregular sampling triggered window resets, false off-state insertions, and incomplete sample counts.

These mechanisms fail toward the negative class, maintaining 100% precision but missing 86% of true blinking events. This is a conservative failure mode rather than an erratic one, which is preferable for a search-and-rescue application, but it renders the capability unusable as deployed.

The implication for future work is that the bottleneck is throughput stability rather than algorithmic design. Reducing per-frame cost, either by invoking the second-stage crop model at a reduced duty cycle, or by lowering inference resolution, or by confirming that inference is genuinely GPU-accelerated on the VOXL2, would raise the sampling rate. Equally important, bounding the *variance* of that rate would allow the existing window logic to function without modification.

3) Window Continuity as a Design Constraint

The temporal stage requires an uninterrupted sequence of detections — four detections across four seconds, as specified in Section H-C. Gaps longer than 5 s insert synthetic off-states (H.16); gaps exceeding 12 s resets the window entirely.

A beacon is hardest to detect during its off-phase, when there is no emitted light and the target is reduced to a low-contrast housing. The window logic is therefore most likely to be interrupted by the very state it exists to measure. Stationary testing succeeded because low sample variance (237 ms) kept detections within timing thresholds. In flight, large timing variations frequently caused window timeouts, discarding valid preliminary detections.

Future iterations must retain partial evidence across brief dropped frames rather than resetting state history.

4) Geolocation and the Ground-Truth Gap

Geolocation was the least mature component of the system. The intended sensor was a ZED stereo camera, which would have supplied range directly from disparity; as noted in Section H-C, the Starling 2 did not meet the SDK's minimum requirements, and the onboard monocular camera was used instead. A single lens cannot recover absolute range without an additional assumption, so the problem became one of selecting which assumption to adopt.

The chosen validation strategy — ArUco fiducials [10] — proved unworkable at operating altitude for a reason visible in Fig. H.2(b). A marker from the 4×4 dictionary requires a 6×6 cell grid to be resolved, including its border, and reliable decoding requires several pixels per cell. At the altitudes flown, the markers resolved to roughly two to three pixels per cell and the oblique viewing angle further foreshortened one axis. This goes below the threshold for decoding. Consequently, the ground-truth fields of our logged data remained empty, and the 2 m geolocation target remains unmeasured.

The fallback approach — estimating range from the known physical dimensions of the beacon and its bounding-box extent — has been implemented, but is not yet validated in the field. as mentioned in Section H-C. Unlike the ArUco method, it assumes that the detected bounding box tightly and consistently bounds the physical beacon, tthe beacon presents a consistent aspect to the camera, and the camera intrinsics in the configuration file are accurate. The first of these is weakest during the off-phase, when the bounding box is driven by the housing rather than the light, which suggests that range estimates and color estimates degrade together rather than independently.

5) Strengths and Limitations

Decoupling perception into a spatial and temporal stage successfully isolated noise while remaining model-agnostic. However, this decomposition makes the temporal stage wholly dependent on the throughput and continuity of the spatial stage.

Two scope limitations of the evaluation itself should be stated plainly. First, the flight results in Section H-D characterize the perception pipeline under a single, deliberately constrained waypoint mission. The results therefore should not be read as evidence about performance across the altitudes, ranges, or trajectories that a deployed search mission would involve. Second, all testing was conducted over land in calm conditions; open-water effects such as glare, surface reflection, and wave-induced motion remain remain untested. Although preliminary testing was conducted in several environments, including over water, the final evaluation was limited to one trial for most beacon configurations. Therefore, the reported results should be interpreted as a preliminary demonstration.

Several conditions would break the approach as currently built. Any increase in per-frame cost, whether from higher resolution or additional model stages, directly reduces blink recall. Beacons blinking faster than approximately 0.5 Hz would fall further outside the achievable sampling rate. Extended detection dropout discards accumulated window history under the 12 s rule. Finally, the entire color stage assumes a beacon that is emitting; a beacon with a failed or occluded LED is

indistinguishable from background clutter to this pipeline.

The riskiest assumption underlying this project was that a perception pipeline validated on a stationary, resource-rich configuration would transfer to a moving, compute-constrained one with degradation proportionate to the change in conditions. Every design decision was tuned to these specific environmental configurations.

An attempt was made to test this assumption through six identical ground-truth conditions. The assumption held for color classification and failed for blink detection.

One assumption remains genuinely untested. The bounding-box range estimator was implemented after the flight campaign and has not been evaluated against any independent reference, precisely because the ArUco method intended to supply that reference failed. Any geolocation accuracy figure derived from the current system should be treated as unsubstantiated until that validation is performed.

F. Conclusion

This work presented an autonomous aerial scouting system for maritime search and rescue with a two-stage perception pipeline for color and blinking detection as well as geographic position estimation with on-board monocular vision. Models were trained in simulation and fine-tuned on real footage from two aircraft, and the complete pipeline was deployed on-board a ModalAI Starling 2 Max with inference performed on-board the aircraft.

We evaluated our algorithms against six ground-truth beacon configurations on two platforms, stationary tests achieved a macro color accuracy of 98.5% and 83.4% during waypoint flights. Blink detection reached 92.7% accuracy in the stationary configuration but fell to 44.2% in flight, with recall of 13.7% against a precision of 100%. Field evaluation also revealed that visual fiducials are unviable for aerial monocular range validation without large target boards or ground GPS references.

Future work focuses on three key areas: optimizing on-board processing throughput on the VOXL2, relaxing rigid window contiguity constraints to handle irregular frame rates, and conducting full field evaluations in open-water environments.

Acknowledgment

The authors would like to thank faculty advisor Dr. George Sklivanitis, Associate Director of the Center for Connected Autonomy and Artificial Intelligence at Florida Atlantic University, and Marcus Fraguela, high-school student intern who assisted with development of the ArUco marker software.

Claude Sonnet 5 was used to aid with coding, debugging, and technical writing. ChatGPT-5.6 Sol Light and Gemini 3.6 Flash were used to assist with editing and non-technical writing.

Data Availability

The source code and datasets used in this study are publicly available at https://github.com/scruz10FAU/EECS-seaOwlRobotX/tree/main. The auto-labeling script is available at https://github.com/scruz10FAU/autolabel.

References

[1] F. Xiao and Y. J. Lee, "Video object detection with an aligned spatial-temporal memory," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018. [Online]. Available: https://www.cs.ucdavis.edu/~yjlee/projects/eccv2018-stmn.pdf

[2] M. Ahrnbom, M. B. Jensen, K. Åström, M. Nilsson, H. Ardö, and T. Moeslund, "Improving a real-time object detector with compact temporal information," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, 2017, pp. 190–197, doi: 10.1109/ICCVW.2017.31.

[3] I. Lapa, S. Ferreira, C. Mateus, N. Rocha, and M. Rodrigues, "Real-time blink detection as an indicator of computer vision syndrome in real-life settings: An exploratory study," *Int. J. Environ. Res. Public Health*, vol. 20, no. 5, p. 4569, Mar. 2023, doi: 10.3390/ijerph20054569.

[4] Y. Zhou, L. Yan, Y. Han, L. Zhu, Y. Zhao, and H. Xie, "HFCH: Hybrid frontier guided fast UAV autonomous exploration for complete and high-quality mapping in unknown environment," *Adv. Eng. Informatics*, vol. 69, p. 103834, Sep. 2025, doi: 10.1016/j.aei.2025.103834.

[5] Exyn, "ExynAI, modular autonomy for mission critical data," May 2022. [Online]. Available: https://www.exyn.com/drones-for-mapping-gps-denied-areas

[6] "Overcoming the sim-to-real gap: Leveraging simulation to learn to explore for real-world RL," *NeurIPS Virtual*, 2024. [Online]. Available: https://neurips.cc/virtual/2024/poster/95701

[7] A. M. Ali, A. Gupta, and H. A. Hashim, "Deep reinforcement learning for sim-to-real policy transfer of VTOL-UAVs offshore docking operations," *Appl. Soft Comput.*, vol. 162, p. 111843, Jun. 2024, doi: 10.1016/j.asoc.2024.111843.

[8] West Marine, "Reading the water: Guide to navigational buoys," 2026. [Online]. Available: https://www.westmarine.com/west-advisor/guide-to-navigational-buoys.html

[9] D. C. Schedl, I. Kurmi, and O. Bimber, "An autonomous drone for search and rescue in forests using airborne optical sectioning," *Sci. Robot.*, vol. 6, no. 55, Jun. 2021, doi: 10.1126/scirobotics.abg1188.

[10] OpenCV, "Detection of ArUco markers," *OpenCV 4.5.0 Documentation*. [Online]. Available: https://docs.opencv.org/4.5.0/d5/dae/tutorial_aruco_detection.html. [Accessed: Jul. 1, 2026].

Appendix I
AirVLA and Screw Gear Style Aerial Grippers

Samantha Garcia
Center for Connected Autonomy & Artificial Intelligence
College of Electrical Engineering & Computer Science
Florida Atlantic University
Boca Raton, FL, USA
scruz10@fau.edu

A. Design Overview

Lightweight, servo-actuated gripper was developed for mounting beneath the ModalAI Starling 2 Max UAV. The gripper is intended for dynamic aerial grasping of small objects approximately 65 mm in diameter, 25 mm tall, and 25–30 g in mass.

Two different mechanisms were designed to lift the object. The first uses two spring-loaded, UMI-style fingers driven symmetrically by a single micro servo. A shared cable-and-pulley system moves the fingers along miniature linear rails, allowing the gripper to open and close around the target object. The estimated overall weight of this gripper was 192.78 grams.

The second mechanism was designed to be much more light-weight which was more conducive to the maximum take-off mass (MTOM) alotted by the Starling 2 Max specifications. This used screw gears, with one in the center for the servo and

one on each of three fingers. The estimated overall weight of this gripper was 40.67 grams.

B. UMI-style Gripper

1) Design Origin and Motivation

The mechanical design for the UMI-style mechanism was derived from the open-source actuated UMI gripper, which was itself based on the Universal Manipulation Interface (UMI) gripper [2]. The adaptation for aerial use follows the approach presented in the AirVLA work, "π, But Make It Fly," in which a lightweight UMI-derived gripper was installed on a ModalAI Starling 2 Max. [1].

This design was selected for the following reasons:

- **Proven end-effector geometry:** The original UMI finger and fingertip geometry was preserved to maintain similarity to previously demonstrated robotic manipulation systems.
- **Established actuation mechanism:** The constant-force-spring return, linear-rail, and cable-and-pulley mechanism had already been demonstrated on handheld and arm-mounted UMI grippers.
- **Reduced development time:** Beginning with an open-source design allowed the development effort to focus on weight reduction, object-specific scaling, and integration with the UAV rather than developing an entirely new mechanism.

2) Aerial Redesign

The original actuated UMI gripper used a DYNAMIXEL XL430-W250-T servo, a ROBOTIS U2D2 interface, a Power Hub, and a mounting interface designed for a Franka Emika robotic arm. These components were not suitable for the UAV installation because of their weight, power requirements, control hardware, and incompatible mounting geometry.

The following modifications were made.

a) Removal of Robot-Arm Hardware

The Franka arm-flange mount, DYNAMIXEL servo, U2D2 interface, and ROBOTIS Power Hub were removed. A new mounting interface was designed using the official Starling 2 Max CAD geometry so that the gripper could be attached directly under the aircraft.

b) Servo Selection and Travel Scaling

The actuator selection progressed through three configurations:

1) the original DYNAMIXEL XL430-W250-T servo;
2) a Hitec HS-785HB/HS-788HB winch servo, selected when the required cable travel was estimated to be approximately 80 mm; and
3) a 9-g-class micro servo, represented by the SG90 or FS90MG, selected after the mechanism was rescaled for the actual target object.

Scaling the opening span for the 65-mm target reduced the required cable travel to approximately 18 mm. This corresponds to approximately 100° of pulley rotation and is therefore within the nominal 180° range of a conventional micro servo.

c) Linear-Rail Reduction

The original 80-mm MGN7 linear rails were shortened to approximately 45–50 mm. The shorter rails provide the required finger travel while reducing the gripper's mass and installed volume.

d) Torque Margin

Two McMaster-Carr 9293K111 constant-force springs provide the return force. The calculated worst-case pulley torque is approximately

$$\tau_{\mathrm{req}} \approx 0.35 \ \mathrm{kgf \cdot cm}. \tag{I.1}$$

The selected micro servo has a rated torque of approximately 1.8–2.2 $\mathrm{kgf \cdot cm}$. The corresponding nominal torque safety-factor range is

$$\mathrm{SF}_\tau = \frac{\tau_{\mathrm{servo}}}{\tau_{\mathrm{req}}} \approx \frac{1.8\text{–}2.2}{0.35} = 5.1\text{–}6.3. \tag{I.2}$$

This calculation indicates that the selected servo provides sufficient static torque margin for the spring-return mechanism. The final assembly must nevertheless be tested for friction, cable tension, transient loading, and servo-voltage effects.

The robotic arm drawing in Figure I.1 shows the plans for the robotic arm.

C. Screw Gear Gripper

1) Design Origin and Motivation

The mechanical design for the Screw Gear mechanism was derived from an open-source servo gripper designed for an SG90 servo [3]. The adaptation involved using thicker wall for durability and integrating the same mounting mechanism as the UMI design.

This design was selected for the following reasons:

- **Proven end-effector geometry:** The original fingers were shown lifting objects similar to those being used in the competition, with fingers that flex around the object to obtain the appropriate grip.
- **Established actuation mechanism:** The screw gear mechanism had already been demonstrated in video included on the open-source gripper page.
- **Reduced development time:** Beginning with an open-source design allowed the development effort to focus on weight reduction, object-specific scaling, and integration with the UAV rather than developing an entirely new mechanism.

2) Aerial Redesign

The original servo gripper demonstrated success, however it required modification in order to be able to mount it to the drone. The walls on the original design were also exceedingly thin and inappropriate for long-term durability. The following modifications were made.

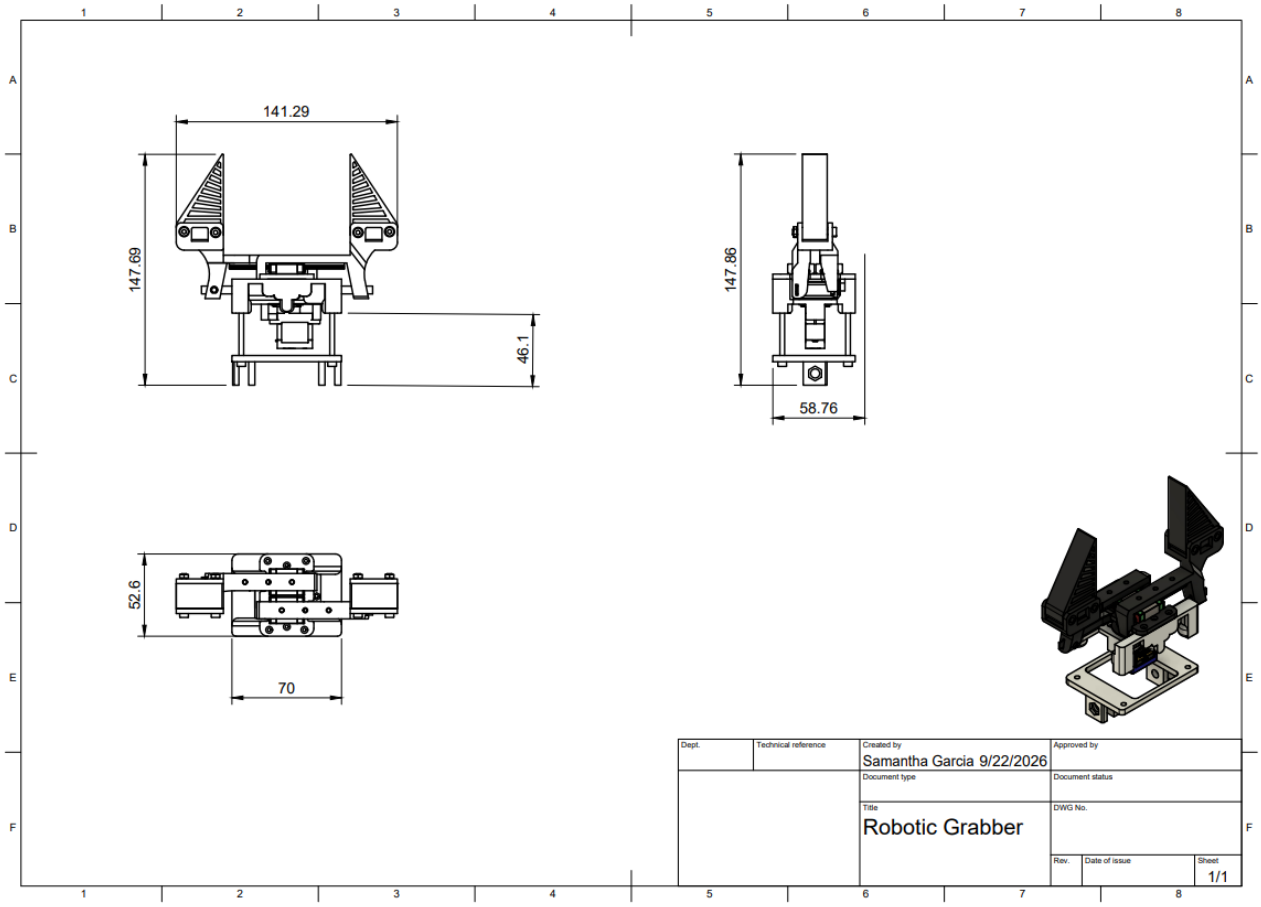


Fig. I.1. UMI robotic arm drawing.

a) Mounting Modification

The base of the gripper was modified to include mounting gear for the UAV. The body used to attach the servo was extended to allow for three additional mounting holes for M3 inserts. These were designed for M3 nylon spacers that would attach to the top of the mount. The top mount was designed to attach to the previously designed UAV mount.

b) Wall Thickness

The original 1mm walls were too thin and not able to withstand minor stress. To ensure long-term performance, the walls were thickened to a minimum of 2mm thickness. Due to the change, other parts needed to be slightly modified to fit the thicker parts. The servo gear was extended at the servo attachment to accommodate the thicker wall on the gripper base. The plastic bolts that hold the gripper fingers were also lengthened.

c) Servo Selection

The target tin measures approximately 2.5 in (63.5 mm) in diameter and 1 in (25.4 mm) in height. These dimensions closely match the gripper's design target of approximately 65 mm in diameter and 25 mm in height. Therefore, the tin fits within the intended grasping envelope without requiring additional finger travel.

The calculated torque required to overcome the two constant-force return springs is approximately

$$\tau_{\text{req}} \approx 0.35 \text{ kgf} \cdot \text{cm}. \tag{I.3}$$

The FS90MG servo provides approximately 1.8–2.2 $\text{kgf} \cdot \text{cm}$ of torque. The resulting actuation torque safety factor is

$$\text{SF}_\tau = \frac{\tau_{\text{servo}}}{\tau_{\text{req}}} = \frac{1.8\text{–}2.2}{0.35} \approx 5.1\text{–}6.3. \tag{I.4}$$

This margin indicates that the servo should have sufficient torque to operate the spring-loaded gripper despite friction and moderate manufacturing variation. However, the torque safety factor describes mechanism actuation and does not directly represent payload-retention capacity. The tin is retained primarily by friction between the compliant TPU fingertips and its outer surface.

The robotic arm drawing in figure I.2 shows the plans for the robotic arm.

D. Aircraft Integration and Clearance

The replacement mounting interface was developed against the Starling 2 Max STEP model. The installed gripper was po-

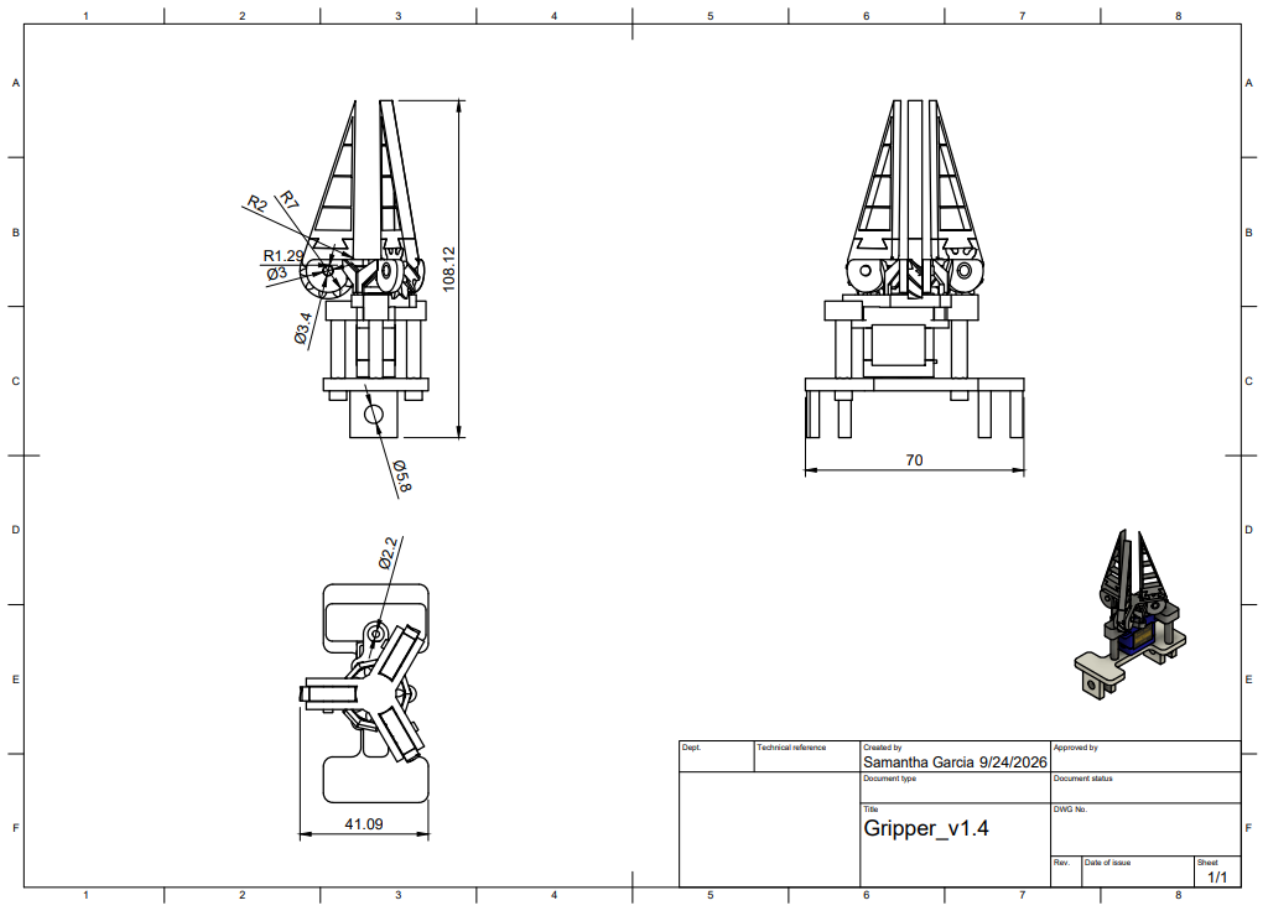


Fig. I.2. Screw gear robotic gripper drawing.

sitioned so that it did not obstruct the aircraft's forward-facing visual-inertial odometry and time-of-flight camera cluster.

The clearance around the separate downward-facing sensor board must be verified on the physical aircraft before flight. The sensor can be checked using the following VOXL command:

```
voxl-inspect-tof
```

E. Materials

The structural components were divided between rigid and compliant materials according to their functions:

- **PETG:** Used for the main chassis, structural body, pulley, mounting components, and rail-adjacent parts. PETG provides adequate strength and wear resistance while remaining accessible to conventional fused-filament fabrication equipment.
- **TPU:** Used for the fingers and flexible drive belt. TPU provides compliant gripping surfaces and can repeatedly bend around the pulley without the brittle failure associated with more rigid printing materials.

Table I.1 lists the components required to construct the aerial grippers.

F. Remaining Verification

Before flight, the completed assembly should be checked to confirm:

- symmetric finger motion throughout the full opening range;
- adequate cable tension without binding or derailment;
- sufficient grip retention for the 25–30-g target payload;
- unobstructed operation of the forward and downward sensors;
- adequate ground clearance during takeoff and landing; and
- secure attachment of all fasteners under aircraft vibration.

The gripper can be observed as it is attached to the bottom of the UAV with inflatable landing gear in Figure I.3.

References

[1] J. Tucker, D. Liu, A. Swann, A. Ren, J. Yu, J. Sun, B. Kim, L. McGranahan, Q. Vuong, and M. Schwager, "π, But Make It Fly: Physics-Guided Transfer of VLA Models to Aerial Manipulation," *arXiv preprint arXiv:2603.25038*, 2026, doi: 10.48550/arXiv.2603.25038. [Online]. Available: https://arxiv.org/html/2603.25038v1. [Accessed: Sep. 24, 2026].

[2] E. Helmut, N. Funk, T. Schneider, and J. Peters, "Actuated Version of the Universal Manipulation Interface Gripper," 2024. [Online]. Available: https://github.com/actuated-umi/actuated-umi-gripper. [Accessed: Sep. 24, 2026].

TABLE I.1
UMI-STYLE AERIAL GRIPPER BILL OF MATERIALS.

#	Item	Qty.	Gripper	Specification	Source	Unit Cost
1	FS90MG micro servo	1	Both	13 g, 180°, 2.2 kgf · cm peak torque, 20-tooth spline	Amazon	$7.10
2	MGN7 linear rail	2	UMI	7-mm rail width; approximately 45 mm long after shortening	Amazon	$13.99
3	MGN7C carriage	1	UMI	Standard MGN7-compatible carriage retained from the existing design	Existing BOM	$0.00
4	Constant-force spring	2	UMI	McMaster-Carr 9293K111; 0.37-lbf rated load; 12-in length	McMaster-Carr	$7.34
5	M3 nylon standoffs kit	1	Both	260pcs M3 Nylon Standoffs Kit; 16-20mm length; male and female	Amazon	$9.99
6	Heat-set insert kit	1	Both	M2–M6 brass inserts, bolts, nuts, and washers; 716 pieces; 60-W installation iron with five tips	Amazon	$9.99
7	PETG filament	1	Both	1.75-mm diameter; 1-kg spool; black	Amazon	$13.29
8	90A TPU filament	1	Both	1.75-mm diameter; 1-kg spool; black	Amazon	$24.99
9	M5 nyloc nuts	1	Both	Nylon-insert lock nuts for vibration-resistant joints	Amazon	$9.99
10	M5×75-mm bolt	1	Both	Fastener for attaching the gripper assembly to the UAV	McMaster-Carr	$6.83
11	M2 flat-head screw kit	1	UMI	Assorted M2 flat-head screws; 660-piece kit	Amazon	$8.99
12	M3×25-mm nylon hex bolts; kit with 20 pcs	1	UMI	Used for internal joints	Amazon	$4.89
13	260pcs M3 Nylon Standoffs Kit, Male Female	1	Both	Used for mounting	Amazon	$9.99

[3] Eitoldas, "Servo Gripper," *Thingiverse*, Nov. 25, 2024. [Online]. Available: https://www.thingiverse.com/thing:6846196. [Accessed: Sep. 25, 2026].



Fig. I.3. Screw-gear gripper installed beneath the Starling 2 Max with inflatable landing gear.

TABLE J.1
M0173 J6 CONNECTIONS USED FOR SERVO CONTROL.

Pin	Signal	Purpose
1	2.8 V	Low-side logic reference
5	I^2C SCL	I^2C clock
6	I^2C SDA	I^2C data
8	GND	Common ground

APPENDIX J
SG90 SERVO INTEGRATION USING THE VOXL 2 I^2C INTERFACE

Samantha Garcia
Center for Connected Autonomy & Artificial Intelligence
College of Electrical Engineering & Computer Science
Florida Atlantic University
Boca Raton, FL, USA
scruz10@fau.edu

This appendix summarizes the hardware integration approach developed for controlling an SG90 micro servo from the ModalAI Starling 2 Max. The servo is intended for use with the UAV-mounted gripper and requires a standard three-wire interface consisting of power, ground, and a pulse-width modulation (PWM) control signal.

The existing Starling 2 Max configuration already utilizes the available M0125 USB/UART breakout board for the RFD900ux telemetry radio and Wi-Fi adapter. Therefore, an alternative interface was selected to avoid disturbing the existing communications hardware. The selected approach uses the I^2C interface exposed through the M0173 front-end board.

A. Available I^2C Interface

The M0173 front-end board exposes an accessory interface through connector J6. In the current UAV configuration, J6 is connected to the Lepton thermal imaging subsystem; however, the thermal camera is not required for the competition configuration. This allows the J6 interface to be repurposed for servo control.

The relevant J6 connections are summarized in Table J.1.

The I^2C interface is accessible from the VOXL 2 application processor as `/dev/i2c-4`. The J6 I^2C signals operate at approximately 2.8 V logic levels.

B. PWM Generation

The SG90 cannot be controlled directly from I^2C because it requires a conventional servo PWM signal. Therefore, an external PCA9685 PWM controller was selected to convert I^2C commands from the VOXL 2 into a servo-compatible PWM output.

A compact PCA9685-based controller was considered to reduce payload size and mass. The RAK13004 WisBlock PWM Expander was identified as one compact option; however, because the RAK13004 is designed to interface with a WisBlock baseboard, a conventional standalone PCA9685 breakout may simplify integration.

Because the M0173 operates at approximately 2.8 V while typical PCA9685 modules operate at 3.3 V logic, a bidirectional I^2C level shifter is included between the M0173 and the PCA9685. A BSS138-based bidirectional level translator is suitable for the open-drain signaling used by I^2C.

C. Servo Power

The SG90 requires approximately 5 V and should not be powered from the low-current logic supply exposed on the M0173. Instead, a dedicated 5 V UBEC is used to provide servo power. A 5 V, 3 A switching UBEC was selected to provide sufficient transient-current capability during servo startup, rapid motion, or stall conditions.

The UBEC input is connected to the UAV battery-side power bus rather than the regulated 5 V VOXL 2 input. This electrically isolates servo current transients from the VOXL 2 flight computer power supply.

The SG90 wiring is:

- Red: regulated 5 V from the UBEC,
- Brown: system ground,
- Orange: PWM output from the PCA9685.

All grounds, including the UBEC, PCA9685, level shifter, M0173, and servo ground, are connected together.

D. System Architecture

The resulting servo-control architecture and wiring is shown conceptually in figure J.1.

E. VOXL 2 Software Configuration

The I^2C bus should first be verified on the VOXL 2 using:

```
ls -l /dev/i2c-4
```

If the I^2C utilities are installed, connected devices can be identified using:

```
i2cdetect -y 4
```

A conventional PCA9685 typically appears at I^2C address `0x40`. The bus should be scanned both before and after connecting the PWM controller to identify existing peripherals and verify that no address conflicts are present.

Servo control may then be implemented directly from Python or another application running on the VOXL 2 application processor. This approach does not require PX4 actuator configuration because the PWM controller is operated through the Linux I^2C interface.

For the SG90, the PCA9685 should initially be configured for approximately 50 Hz output. A conservative initial pulse-width range of approximately 1200–1800 μs is recommended during mechanical testing, with approximately 1500 μs corresponding to the center position. The range can subsequently be expanded after confirming that the gripper mechanism does not contact a mechanical hard stop.

F. Integration Considerations

Before permanently disconnecting the existing J6 cable, the downward rangefinder functionality should be verified. The M0173 J6 interface may also support the Starling's rangefinder subsystem in addition to the unused Lepton thermal camera. If disconnecting J6 disables a required rangefinder, the I^2C bus should instead be shared between the existing sensor and the PCA9685, provided that their I^2C addresses do not conflict.

This approach was selected because it preserves the existing RFD900ux telemetry and Wi-Fi interfaces, avoids adding the larger M0065 VOXL 2 IO PWM expansion hardware, and provides a compact method for controlling the single SG90 gripper actuator directly from the VOXL 2 application processor.

References

[1] ModalAI, "M0173 VOXL 2 Starling 2 Front End," ModalAI Documentation. [Online]. Available: https://docs.modalai.com/M0173/ [Accessed: Sep. 28, 2026].

[2] ModalAI, "Rangefinders – How to Use VL53L1CX Rangefinders with VOXL 2," ModalAI Documentation. [Online]. Available: https://docs.modalai.com/rangefinders/ [Accessed: Sep. 28, 2026].

[3] NXP Semiconductors, "PCA9685: 16-Channel, 12-Bit PWM Fm+ I2C-Bus LED Controller," PCA9685 Data Sheet, Rev. 4.0, Apr. 2015. [Online]. Available: https://www.nxp.com/products/power-drivers/lighting-driver-and-controller-ics/led-drivers/16-channel-12-bit-pwm-fm-plus-ic-bus-led-driver:PCA9685 [Accessed: Sep. 28, 2026].

[4] RAKwireless, "RAK13004 PCA9685 PWM Expansion Module." [Online]. Available: https://store.rakwireless.com/products/pwm-expansion-module-rak13004 [Accessed: Sep. 28, 2026].

[5] Adafruit Industries, "4-Channel I2C-Safe Bi-Directional Logic Level Converter – BSS138," Product ID 757. [Online]. Available: https://www.adafruit.com/product/757 [Accessed: Sep. 28, 2026].

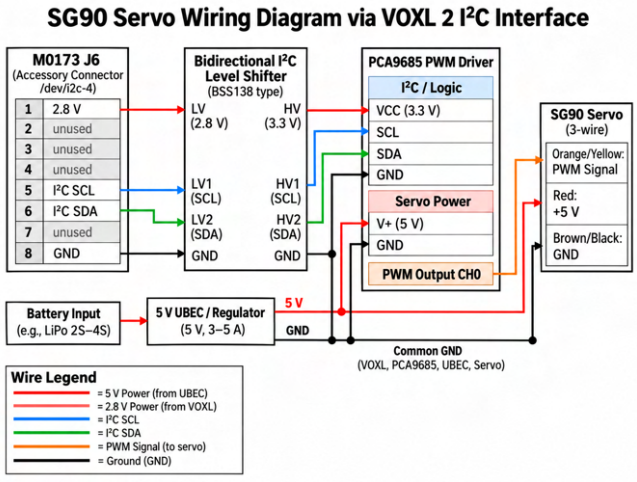
[6] Tower Pro, "SG90 Analog Micro Servo." [Online]. Available: https://towerpro.com.tw/product/sg90-analog/ [Accessed: Sep. 28, 2026].

Appendix K
ELRS 2.4 GHz MAVLink Telemetry Link Setup

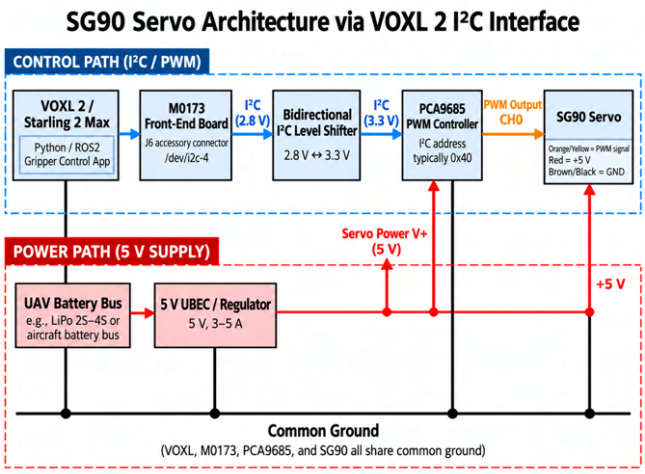
Clarck Dorcent
Center for Connected Autonomy & Artificial Intelligence
College of Electrical Engineering & Computer Science
Florida Atlantic University
Boca Raton, FL, USA
cdorcent2021@fau.edu

A. System Configuration and Objective

Aircraft: ModalAI Starling 2 Max with VOXL 2 and PX4 on the Snapdragon Flight platform.
Ground equipment: Jumper AION ELRS 2.4-GHz Nano transmitter module with a Diamond antenna, connected to the QGroundControl (QGC) laptop through USB-C [3].
Airborne equipment: RadioMaster RP1-V2 receiver used as a second, telemetry-only unit connected to a VOXL 2 Linux UART.
Objective: Establish a MAVLink telemetry connection between QGC and the aircraft that remains separate from the safety pilot's RC-control link. ExpressLRS supports bidirectional MAVLink and direct serial connection to a operator control station [1].



(a) Servo Wiring



(b) Servo Architecture

Fig. J.1. Servo wiring and architecture.

The phases below shall be completed in order. Bench testing shall be completed before the equipment is installed or tested on the aircraft, and the existing VOXL configuration shall be backed up before changes are made.

Declaration note: This configuration corresponds to the declared telemetry allocation of 2404–2479 MHz at 10 mW EIRP. Appendix B, Section B5 of the compliance documentation must be updated.

B. Phase 0: Back Up the VOXL

With the aircraft powered and connected through USB, run:

```
adb shell
mkdir -p /data/backup_$(date +%F)
cp /etc/modalai/voxl-mavlink-server.conf /data/
    backup_$(date +%F)/
systemctl cat rfd900-mavlink
```

Complete the following backup actions:

- Use the final command to identify the service file and bridge script path.
- Copy the service file and `rfd900_mavlink_bridge.py` into the backup directory, then transfer them to the laptop with `adb pull`.
- Export the PX4 parameters from QGC over Wi-Fi.
- Open a new MOD entry in the aircraft build log before changing the configuration.

C. Phase 1: Flash the AION

Complete this phase using the laptop only; do not connect the aircraft. The ExpressLRS AION update procedure is documented in [2].

1) Install the Diamond antenna before applying power. Never power an ELRS transmitter module without its antenna installed.
2) Install ExpressLRS Configurator from the project's official releases page.
3) Connect the AION to the laptop through USB-C. Run `dmesg | tail` and record the assigned port, such as `/dev/ttyUSB0` or `/dev/ttyACM0`.
4) In ExpressLRS Configurator, use the following settings:
   - Target: Jumper 2.4 GHz → Jumper AION Nano 2.4 GHz TX;
   - flashing method: UART;
   - firmware: the current compatible 3.x release, version 3.5.0 or newer;
   - regulatory domain appropriate for the operating location; and
   - a new binding phrase that is different from the binding phrase used by the pilot's RC link.
5) Flash the module and verify the installed version on the AION display.

D. Phase 2: Flash the RP1

Complete this phase using a regulated bench supply rather than aircraft power.

1) Supply 5 V to the RP1 and verify the voltage with a multimeter before connecting the receiver.
2) Allow the receiver to enter Wi-Fi mode if it does not bind.
3) Connect the laptop to the receiver's `ExpressLRS RX` Wi-Fi network.
4) In ExpressLRS Configurator, select the RadioMaster RP1 target matching the installed board and confirm that it identifies the V2 hardware. Select Wi-Fi flashing, the same ExpressLRS version as the AION, and the same new binding phrase.
5) Flash the receiver.

E. Phase 3: Perform the Bench Link Test

1) Connect the AION to the laptop through USB and power the RP1 from the regulated 5-V bench supply.
2) Set the AION output power to 10 mW.

TABLE K.1
RP1-TO-VOXL UART CONNECTIONS.

RP1 Connection	VOXL Connection
TX	J3 RX
RX	J3 TX
GND	GND
5 V	Verified regulated 5-V source

3) Confirm that the connected indicator appears on the AION display.
4) Determine whether the AION maintains the link while powered only through USB and without a handset:
   - If the connection remains stable, continue to Phase 4.
   - If the AION does not connect or repeatedly reboots, restart with compatible handset or separate 5–8.4-V supply is required.

F. Phase 4: Enable MAVLink Mode

ExpressLRS requires compatible transmitter and receiver firmware for MAVLink operation and forces the telemetry ratio to 1:2 while MAVLink mode is active [1].

1) With the AION and RP1 linked, open *Other Devices* in the AION menu and set the RP1 serial protocol to MAVLink.
2) Power off the RP1.
3) On the AION, change *Link Mode* from *Normal* to *MAVLink*.
4) Restore power to the RP1 and confirm that the devices reconnect.

G. Phase 5: Connect the RP1 to the VOXL

Perform all wiring with aircraft power removed. Use the official VOXL 2 connector documentation to verify the M0151 J3 pinout; pin assignments shall not be inferred or guessed [4].

1) Before removing aircraft power, run the following command and confirm that `/dev/ttyHS1` exists:

```
adb shell ls /dev/ttyHS*
```

   Confirm that the `rfd900-mavlink` service is disabled.
2) Power off the aircraft.
3) Obtain and verify the M0151 J3 connector pinout.
4) Make the UART connections shown in Table K.1.

Before connecting the RP1, temporarily restore aircraft power and verify 5 V at the selected power pin and 3.3-V logic levels on the UART lines. Remove power again before connecting the receiver. Route the RP1 antenna away from the GPS antenna, Wi-Fi adapter, and primary RC receiver.

H. Phase 6: Configure the VOXL Bridge

The existing VOXL MAVLink service configuration should be retained as a backup. ModalAI documents the operation and configuration of `voxl-mavlink-server` in [5].

1) Copy the existing bridge script to a new file. Do not edit the original copy.
2) Change the serial baud rate from 57600 to 460800.
3) Add a filter that prevents MAVLink message 70, `RC_CHANNELS_OVERRIDE`, from being forwarded to PX4.
4) Run the new bridge manually before creating a service:

```
python3 /path/to/elrs_mavlink_bridge.py
```

5) Create and enable a systemd service only after the connection and safety tests in Phases 7 and 8 pass.

I. Phase 7: Connect QGC Through ELRS

1) Disable either laptop Wi-Fi or aircraft Wi-Fi so that telemetry can reach QGC only through the ELRS link.
2) In QGC, open *Application Settings* → *Comm Links* and disable every auto-connect option.
3) Add a serial link using the AION port and a baud rate of 460800. On Linux, leave *Force DTR Low* unchecked, as required by the ExpressLRS QGC procedure [1].
4) Connect and verify that:
   - the QGC heads-up display updates;
   - aircraft parameters download successfully, recognizing that throughput of approximately 9.9 kbps may make the download slow; and
   - a small test mission uploads successfully.

J. Phase 8: Perform Safety Tests

Remove all propellers before performing these tests.

1) **RC-loss test:** With the RC and telemetry links active, turn off the pilot's handset. QGC must indicate RC loss. If RC loss is not reported, the telemetry link may be providing RC input. Stop the test and correct the MAVLink message filter.
2) **Telemetry-loss test:** Disconnect the AION. QGC should report telemetry loss after approximately 20 s. Complete the related steps in Section 4 of the UAV failsafe test procedure.
3) **Interference test:** Compare the primary RC link-quality value with ELRS telemetry enabled and disabled. Record any reduction in link quality.

K. Phase 9: Update Project Records

1) Add MOD, CFG, and TEST entries for the RP1 wiring, the new bridge configuration, and each completed test.
2) Update the telemetry entry in the competition compliance document.
3) Update Appendix B, Section B5 if it still identifies the RFD900ux.
4) Submit the required radio-change notice to the competition governing body.

L. Open Verification Items

The following items must be resolved and recorded during implementation:

- whether the AION can maintain the link using laptop USB power alone;
- the verified M0151 J3 pinout and the selected regulated 5-V source for the RP1;
- whether the RP1 sends RC-override messages while operating in MAVLink mode and whether PX4 on VOXL 2 interprets those messages as RC input; and
- how the existing bridge parses MAVLink packets so that message 70 can be filtered without affecting other traffic.

REFERENCES

[1] ExpressLRS Project, "MAVLink," *ExpressLRS Documentation*. [Online]. Available: https://www.expresslrs.org/software/mavlink/. [Accessed: Sep. 25, 2026].

[2] ExpressLRS Project, "Jumper Aion Nano," *ExpressLRS Documentation*. [Online]. Available: https://www.expresslrs.org/quick-start/transmitters/jumper-aion/. [Accessed: Sep. 25, 2026].

[3] Jumper Technology, "Jumper 2.4G ELRS TX Module," product webpage. [Online]. Available: https://www.jumper-rc.com/modules/jumper-2-4g-elrs-tx-module/. [Accessed: Sep. 25, 2026].

[4] ModalAI, Inc., "VOXL 2 Connectors," *ModalAI Technical Documentation*. [Online]. Available: https://docs.modalai.com/voxl2-connectors/. [Accessed: Sep. 25, 2026].

[5] ModalAI, Inc., "voxl-mavlink-server," *ModalAI Technical Documentation*. [Online]. Available: https://docs.modalai.com/voxl-mavlink-server/. [Accessed: Sep. 25, 2026].

APPENDIX L
ACRONYMS

TABLE L.1: Acronyms

Acronym	Definition
AI	Artificial Intelligence
APF	Artificial Potential Field
ASV	Autonomous Surface Vehicle
BMS	Battery Management System
CAAI	Center for Connected Autonomy and Artificial Intelligence
CMA-ES	Covariance Matrix Adaptation Evolution Strategy
COTS	Commercial Off the Shelf
CV	Computer Vision
DL	Deep Learning
DRL	Deep Reinforcement Learning
ELRS	Express Long Range System
ESC	Electronic Speed Controller
FAA	Federal Aviation Administration
FAU	Florida Atlantic University
GNC	Guidance, Navigation, and Control
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GUI	Graphical User Interface
LiDAR	Light Detection and Ranging
Li-Ion	Lithium Ion
MAVLink	Micro Air Vehicle Link
MAVROS	Micro Air Vehicle Link Robot Operating System
MP	Megapixel
MPA	Modal Pipe Architecture
NDAA	National Defense Authorization Act
NOTAM	Notice to Airmen
OCS	Operator control station PCB
Printed Circuit Board	
PID	Proportional-Integral-Derivative
PLA	Polylactic Acid
PoE	Power over Ethernet
PPO	Proximal Policy Optimization
PWM	Pulse Width Modulation
PX4	Pixhawk 4
QGC	QGroundControl RC
Radio Control	
RGB	Red-Green-Blue
ROS2	Robot Operating System 2
RPIC	Remote Pilot in Command
RRT	Rapidly-Exploring Random Tree
TFLite	TensorFlow Lite
ToF	Time-of-Flight
TPU	Thermoplastic Polyurethane
UAV	Unmanned Aerial Vehicle
USB	Universal Serial Bus
VLOS	Visual Line of Sight
VRX	Virtual RobotX
WLAN	Wireless Local Area Network
YOLO	You Only Look Once

APPENDIX M
LIST OF MAJOR ASV COMPONENTS

TABLE M.1: Major ASV components

Component	Vendor	Model/Type	Specs	Custom/Purchase	Cost	Year of Purchase	Reasoning
ASV Computer	NVIDIA	Jetson Orin AGX	32 GB RAM; 2 Tb SSD	Loan from FAU Engineering Department	1999.99	N/A	ASV Vision Processing
ASV Custom PCB	FAU E-Lab	ASV Main Control Board	Teensy 4.1 Microcontroller, 8 PWM out, embedded safety, fused outputs, 6 subsystem communications, computer power control	Custom	520.00	2025	ASV motor Control, Radio communication, subsystem communication and power distribution, RC monitoring, light stack control, safety system monitoring
ASV LiDAR	Velodyne	VLP-16 HighRes	16 Beams, 100 m radius	Loan from FAU Engineering Department	15,000.00	N/A	Sensor for autonomous navigation, short and medium ranged performance
ASV Camera	Amcrest	4k PoE Camera	15 fps 129 DEG FOV	Purchased	399.97	2025	Object classification and outdoor water resistant sensor
ASV GPS	Garmin	P19x HVS	10Hz position, velocity and time data NMEA	Purchased	169.99	2024	ASV Sensors
ASV IMU	Adafruit	BNO055 Absolute Orientation Sensor	100Hz Orientation, Angular Velocity, Acceleration	Purchased	34.95	2025	ASV Sensors
ASV Network Bridge Module	Ubiquiti	RP-5AC-GEN2	5 GHz	Loan from FAU Engineering Department	230.00	N/A	ASV Communications
ASV Light Tower	Banner Engineering Corporation	TL50BLGYRQ	Stack Light Complete Unit Green, Red, Yellow LED Wire Leads	Loan from FAU Engineering Department	263.00	N/A	ASV Safety and Status

Component	Vendor	Model/Type	Specs	Custom/Purchase	Cost	Year of Purchase	Reasoning
ASV Network Edgerouter	Ubiquiti	ER-X-SFP	5-PORT GIGABIT Router w/POE	Purchased	99.00	N/A	ASV Communications
ASV PoE Switch	NETGEAR	5-Port PoE Gigabit Unmanaged Switch	4x PoE; 5x Gigbabit Ethernet Ports; 63W	Purchased	69.99	2025	Networking and power for PoE cameras
ASV Thrusters	Blue Robotics	T-500	16 Kg f thrust	Purchased	952.00	2025	ASV Propulsion
ASV Batteries	Amazon	6S LiPO TATTU 30000 mAh	6S LiPO TATTU 30000 mAh	Purchased	999.99	2025	ASV Propulsion

APPENDIX N
LIST OF MAJOR UAV COMPONENTS

TABLE N.1: Major Starling 2 Max UAV components.

Component	Vendor	Model/Type	Specifications	Custom/Purchase	Cost	Year of Purchase	Reasoning
UAV Platform	ModalAI	Starling 2 Max	Quadrotor UAV; baseline aircraft weight of 566 g; FAA registration FA3WATHAK3	Loan from FAU CAAI	4199	2025	Primary UAV platform for the RobotX competition
UAV Companion Computer and Autopilot	ModalAI	VOXL 2	PX4 v1.14.0; VOXL software v1.5.0; ROS 2 Foxy; integrated flight-control and companion-computing functions	Included with aircraft	Included	2025	Provides flight control, onboard computing, sensor processing, ROS 2 integration, and autonomous mission execution
UAV GPS/GNSS	u-blox	M10 / M10-5883	GNSS receiver with integrated magnetometer interface; connected through J19 GNSS UART and magnetometer I^2C interface	Included and modified	Included	2025	Provides position, velocity, timing, home-position, and heading data for autonomous navigation and RTL operation
UAV Vision Cameras	ModalAI	Dual M0161 and dual M0166 camera modules	Integrated forward, tracking, and downward-facing vision sensors; `hires_front` image stream used by the beacon-detection pipeline	Included with aircraft	Included	2025	Provides imagery for beacon detection, visual perception, and autonomous mission support
Rear Downward Camera Mount	ModalAI	951-A replacement mount	Replacement camera mount purchased in plastic parts kit following damage to the original component	Purchased replacement	42.58	2026	Secures and aligns the rear downward-facing camera after the original mount was damaged during an emergency landing
GPS Mount	ModalAI	GPS replacement mount	GPS mount purchased in plastic parts kit following damage to the original component	Purchased replacement	42.58	2026	Secures and aligns the GPS mount after the original mount was damaged during an emergency landing
UAV Wi-Fi Adapter	ALFA Network	AWUS036EACS [9]	IEEE 802.11a/b/g/n/ac; 2.4 and 5 GHz operation; USB interface; up to 150 Mbps at 2.4 GHz and 433 Mbps at 5 GHz	Included with aircraft	Included	2025	Provides local high-bandwidth communication, software access, mission upload, and QGroundControl connectivity
UAV Telemetry Radio	Amazon	RP1 V2 ELRS	ExpressLRS receiver; configured to MAVLink mode for telemetry	Purchased	18.99	2026	Provides long-range bidirectional MAVLink telemetry between the UAV and UAV ground station while retaining Wi-Fi connectivity
UART Interface Board	ModalAI	M0151 UART add-on board	Provides the J3 UART interface for the RFD900ux communication signals; Linux endpoint `/dev/ttyHS1`	Included / integrated	49.99	2025	Provides the physical UART connection between the RFD900ux telemetry radio and VOXL 2
RC Receiver	Not recorded	RP1 V2 ELRS	ExpressLRS manual-control and safety-pilot receiver; mapped to PX4 RC control inputs	Purchased	18.99	2026	Provides the independent safety-pilot control link, flight-mode selection, arming control, and emergency control inputs
UAV Battery System	Not recorded	Two 2S Li-ion 18650 packs	Two 2S lithium-ion battery packs connected in series through an XT30 connector	Included with aircraft	Included	2025	Provides primary electrical power for the aircraft propulsion, flight-control, computing, and sensor systems
UAV Motors	Not recorded	2203.5 1500 KV brushless motors	Four brushless motors rated at 1500 KV	Included with aircraft	Included	2025	Provides propulsion and attitude control for the quadrotor aircraft
UAV Propellers	Not recorded	180 mm tri-blade propellers	Four 180 mm three-blade propellers configured for the required clockwise and counterclockwise motor rotations	Included with aircraft	Included	2025	Generates lift and control forces for aircraft flight
UAV Landing and Flotation Gear	Custom / FAU	TPU landing gear with pool-noodle flotation	TPU landing-gear structure with direct attachments for 3-in pool-noodle flotation elements; approximately 100 g net increase over the stock configuration	Custom fabricated	30.00	2026	Provides ground clearance, impact absorption, and positive buoyancy for maritime UAV operations

APPENDIX O
UAV TRADE STUDIES

Candidate components and communication methods were evaluated using weighted decision matrices. Each criterion was assigned a weight w_i, and each candidate received a score s_{ij} from 1 (least favorable) to 10 (most favorable). Except where noted, the normalized score for candidate j was calculated as

$$T_j = 100\,\frac{\sum_i w_i s_{ij}}{10\sum_i w_i}. \tag{O.1}$$

The highest numerical result is shown in bold. A high score identifies the best result under the recorded criteria and weights; it does not override mandatory requirements, availability, regulatory compliance, or integration constraints.

A. 20 Ah Battery Trade Study

The 20 Ah battery study weighted mass most heavily, followed by price and discharge rating. The weights were 5, 2, and 1, respectively.

TABLE O.1: Weighted trade study for 20 Ah battery candidates.

Candidate	Price (2)	C Rating (1)	Weight (5)	Total
Lumenier NAV 20000 mAh 6S 21700 Li-ion [1]	6	5	7	65.00%
Tattu Semi-Solid State 20000 mAh 6S 5C LiPo [2]	3	7	7	60.00%
Horus Dynamics LiPo 6S 20000 mAh 20C [3]	2	10	1	23.75%
Turnigy High-Capacity 6S 20000 mAh 12C LiPo [4]	8	9	1	37.50%
MAD 6S 20 Ah Solid-State Li-ion [5]	2	9	9	72.50%
MG UAV 6S 20000 mAh 7C Solid-State [6]	5	8	9	**78.75%**

Result: The MG UAV 6S 20000 mAh battery received the highest weighted score because its favorable mass score dominated the matrix.

TABLE O.2: Supporting data recorded for the 20 Ah battery candidates.

Candidate	Price (USD)	C Rating	Mass (g)
Lumenier NAV 20000 mAh 6S 21700 Li-ion [1]	299.00	1.75	1815
Tattu Semi-Solid State 20000 mAh 6S 5C LiPo [2]	419.00	5	1820
Horus Dynamics LiPo 6S 20000 mAh 20C [3]	410.00	20	2350
Turnigy High-Capacity 6S 20000 mAh 12C LiPo [4]	142.12	12	2630
MAD 6S 20 Ah Solid-State Li-ion [5]	450.00	5	1595
MG UAV 6S 20000 mAh 7C Solid-State [6]	320.00	7	1420
MaxAmps Li-ion 20000 mAh 6S4P, 21.6 V (not scored) [7]	672.00	5	1750

B. 10 Ah Battery Trade Study

The 10 Ah study used the same criterion weights as the 20 Ah study.

TABLE O.3: Weighted trade study for 10 Ah battery candidates.

Candidate	Price (2)	C Rating (1)	Weight (5)	Total
Lumenier NAV 10000 mAh 6S 21700 Li-ion [8]	7	8	8	**77.50%**
MaxAmps 10000 mAh 6S2P Li-ion [9]	5	7	9	**77.50%**

Result: The candidates tied at 77.50%. The supporting data recorded the MaxAmps battery as lighter and NDAA compliant, while the Lumenier battery was less expensive. Therefore, the final selection requires the compliance requirement to be applied as a gate rather than another weighted preference.

TABLE O.4: Supporting data recorded for the 10 Ah battery candidates.

Candidate	Price (USD)	C Rating	Mass (g)	Dimensions (mm)	Sourcing Note
Lumenier NAV 10000 mAh 6S 21700 Li-ion [8]	149.99	9	975	$150 \times 65 \times 55$	U.S. sourced; NDAA status not stated
MaxAmps 10000 mAh 6S2P Li-ion [9]	336.00	5	864	$127 \times 43 \times 75$	Recorded as NDAA compliant

C. Camera Trade Study

The camera study assigned weights of 3 to component capabilities, 1 to price, and 0.5 to accessibility. Recorded requirements included NDAA compliance, a field of view greater than 120°, a global shutter, and the ability to transmit imagery through an onboard Wi-Fi-capable computer.

TABLE O.5: Weighted camera trade study.

Candidate	Components (3)	Price (1)	Access. (0.5)	Total
TechNexion TEVS-AR0234 [10]–[12]	8	10	10	**86.67%**

Candidate	Components (3)	Price (1)	Access. (0.5)	Total
AR0144 monochrome camera module	8	10	10	**86.67%**
TechNexion TEVS-AR0521	7	10	10	80.00%

Result: The TEVS-AR0234 and AR0144 tied. The AR0234 was recorded as a compact NDAA-compliant 1080p color global-shutter camera with an interchangeable M12 lens, but it requires a suitable wide-angle lens and a companion computer for Wi-Fi transmission. The AR0144 offers a recorded 160° field of view and global shutter but is monochrome and limited to 1280×800. The AR0521 provides 1080p color imagery but uses a rolling shutter and therefore does not satisfy the recorded global-shutter requirement.

D. Control-Link Protocol Trade Study

The control-link comparison emphasized bandwidth, with range and compatibility as secondary factors.

TABLE O.6: Weighted control-link protocol trade study.

Protocol	Bandwidth (3)	Range (2)	Compat. (1)	Total
ExpressLRS (ELRS)	10	9	9	**95.00%**
Crossfire (CRSFv3)	5	8	5	60.00%
ImmersionRC Ghost (GHST)	5	9	5	63.33%

Result: ExpressLRS received the highest score at 95.00%. The selected hardware and regional radio settings must still comply with the competition site's spectrum and transmit-power requirements.

E. Flight-Controller Trade Study

The source worksheet compared three controllers, but its decision factors remain labeled only as "Decision Factor 1," "Decision Factor 2," and "Decision Factor 3." The worksheet also stored totals on a 0–10 scale while formatting them as percentages. Table O.7 reports the source scores and normalizes the displayed totals from 770%, 900%, and 830% to 77%, 90%, and 83%, respectively.

TABLE O.7: Flight-controller trade study as recorded in the source worksheet.

Candidate	Factor 1	Factor 2	Factor 3	Normalized Total
Rotor Riot Brave F7 [15]	9	7	8	77%
TBS Lucid H7 (ICM42688) [14]	9	9	8	**90%**
ARK FPV Flight Controller [13]	9	8	7	83%

Result: The TBS Lucid H7 has the highest recorded total. The criterion names, weights, and manually entered weighted values should be corrected in the source study before this table is used as a final design justification.

F. Jetson Wi-Fi Trade Study

The Jetson Wi-Fi comparison emphasized bus compatibility, followed by size and performance.

TABLE O.8: Weighted Jetson Wi-Fi trade study.

Candidate	Bus Type (3)	Size (2)	Performance (1)	Total
Intel AC9260	8	8	8	**80.00%**
ARK version (pending reply)	–	–	–	Not scored
TRENDnet TEW-805UBH AC1200	2	3	5	28.33%

Result: The Intel AC9260 received the highest completed score. The ARK option was incomplete and must not be treated as a zero-score candidate in the final comparison.

G. Telemetry-Radio Trade Study

The telemetry-radio comparison weighted compatibility twice as heavily as range or reliability.

TABLE O.9: Weighted telemetry-radio trade study.

Candidate	Compat. (2)	Range (1)	Reliab. (1)	Total	Integration/Cost Note
DH Research Telemetry Radio [16]	8	7	8	77.50%	UART; USB adapter required
ViuLinx PRO-N [17]	9	10	10	**95.00%**	Ethernet/UART; approximately $2,000 each
RFD900ux/RFD900x [18]	8	10	10	90.00%	UART; approximately $362 per bundle; UX airborne and X ground-side
mRo DualBand Wi-Fi Telemetry Radio [19]	5	4	5	47.50%	Wi-Fi/UART; intended for the ODIN module and potentially harder to integrate without it

Result: ViuLinx PRO-N received the highest weighted score, while the RFD900ux/RFD900x was the next-highest candidate and had a much lower recorded acquisition cost. Cost was not included as a weighted criterion, so the numerical winner does not by itself establish the final selection.

TRADE-STUDY REFERENCES

[1] RaceDayQuads, "Lumenier NAV 20000mAh 6S 21700 lithium-ion battery–XT90," [Online]. Available: https://www.racedayquads.com/products/lumenier-nav-20000mah-6s-21700-lithium-ion-battery-xt90. [Accessed: Sep. 17, 2026].

[2] Tattu, "Tattu semi-solid-state 20000mAh 6S 5C 22.2V LiPo battery pack with XT90-S plug," [Online]. Available: https://www.genstattu.com/tattu-semi-solid-state-20000mah-6s-5c-22-2v-lipo-battery-pack-with-xt90s-plug/. [Accessed: Sep. 17, 2026].

[3] Horus Dynamics, "LiPo 6S 20000 mAh battery," [Online]. Available: https://store.horusdynamics.com/a/battery-lipo-6s-20000-mah-dji-s900-s1000/. [Accessed: Sep. 17, 2026].

[4] HobbyKing, "Turnigy high-capacity 20000mAh 6S 12C drone LiPo pack–XT90," [Online]. Available: https://hobbyking.com/turnigy-high-capacity-battery-20000mah-6s-12c-drone-lipo-pack-xt90.html. [Accessed: Sep. 17, 2026].

[5] UnmannedRC, "MAD 6S 20Ah drone solid-state lithium-ion battery," [Online]. Available: https://unmannedrc.com/products/mad-6s-20ah-drone-solid-state-lithium-ion-battery. [Accessed: Sep. 17, 2026].

[6] UnmannedRC, "MG UAV solid-state 20000mAh 7C battery," [Online]. Available: https://unmannedrc.com/products/mg-uav-solid-state-battery-6s-12s-14s-20000mah-7c-for-heavy-load-long-endurance-uavs. [Accessed: Sep. 17, 2026].

[7] MaxAmps, "Li-ion 20000mAh 6S4P 21.6V battery pack," [Online]. Available: https://maxamps.com/collections/6s-21-6v-li-ion/products/li-ion-20000-6s4p-21-6v-battery-pack. [Accessed: Sep. 17, 2026].

[8] GetFPV, "Lumenier NAV 10000mAh 6S 21700 lithium-ion battery–XT60," [Online]. Available: https://www.getfpv.com/lumenier-nav-10000mah-6s-21700-lithium-ion-battery-xt60.html. [Accessed: Sep. 17, 2026].

[9] MaxAmps, "Li-ion 10000mAh 6S2P 21.6V battery pack," [Online]. Available: https://maxamps.com/products/li-ion-10000-6s2p-21-6v-battery-pack. [Accessed: Sep. 17, 2026].

[10] TechNexion, "TechNexion NDAA-compliant cameras," [Online]. Available: https://www.technexion.com/resources/technexion-ndaa-compliant-cameras/. [Accessed: Sep. 17, 2026].

[11] TechNexion, "TEVS-AR0234 embedded vision camera," [Online]. Available: https://www.technexion.com/products/embedded-vision/image-sensors/tevs-ar0234/. [Accessed: Sep. 17, 2026].

[12] e-con Systems, "AR0234 global-shutter camera module," [Online]. Available: https://www.e-consystems.com/camera-modules/ar0234-global-shutter-camera-module.asp. [Accessed: Sep. 17, 2026].

[13] ARK Electronics, "ARK FPV flight controller," [Online]. Available: https://arkelectron.com/product/ark-fpv-flight-controller/. [Accessed: Sep. 17, 2026].

[14] Ready Made RC, "TBS Lucid H7 flight controller–ICM42688–NDAA," [Online]. Available: https://www.readymaderc.com/products/details/86342-tbs-lucid-h7-flight-controller-icm42688-ndaa. [Accessed: Sep. 17, 2026].

[15] Rotor Riot, "Brave F7 flight controller–Made in USA," [Online]. Available: https://rotorriot.com/products/rotor-riot-brave-f7-flight-controller-made-in-usa. [Accessed: Sep. 17, 2026].

[16] DH Research, "Telemetry Radio V1," [Online]. Available: https://dh-research.com/product/telemerty_radio_v1/. [Accessed: Sep. 17, 2026].

[17] UAV Garage, "Taisync ViuLinx PRO-N," [Online]. Available: https://uavgarage.com/shop/taisync-viulinx-pro-n/. [Accessed: Sep. 17, 2026].

[18] RFDesign, "RFD900ux modem bundle," [Online]. Available: https://store.rfdesign.com.au/rfd900ux-modem-bundle-hs-8517-62-00-90/. [Accessed: Sep. 17, 2026].

[19] mRo, "DualBand Wi-Fi telemetry radio," [Online]. Available: https://mrobotics.io/docs/mro-dualband-wi-fi-telemetry-radio/. [Accessed: Sep. 17, 2026].

APPENDIX P
VOXL 2 ARCHITECTURE

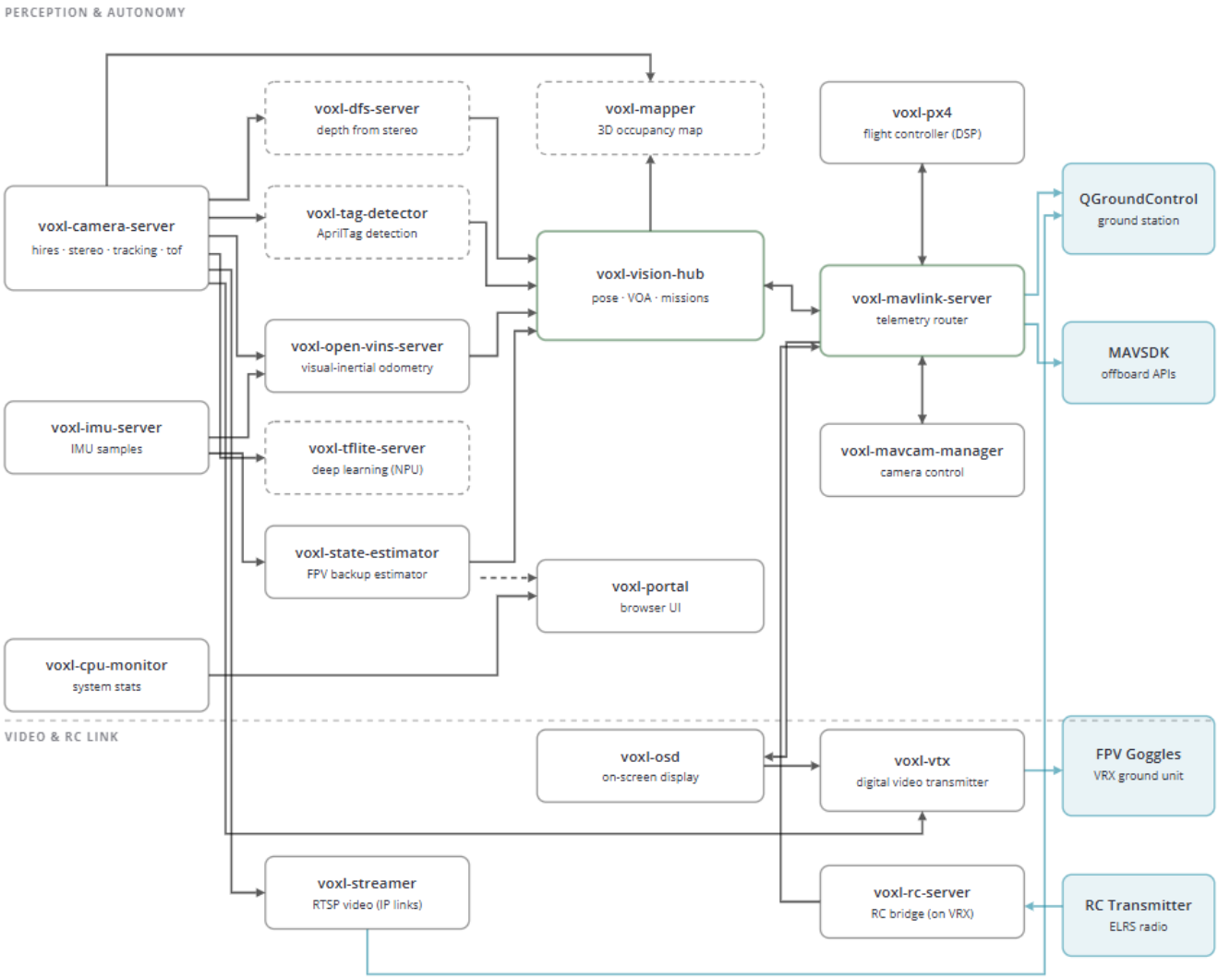


Fig. P.1. VOXL SDK architecture. Reproduced from ModalAI [13].

APPENDIX Q
MODIFIED STARLING 2 MAX UAV MECHANICAL DRAWINGS

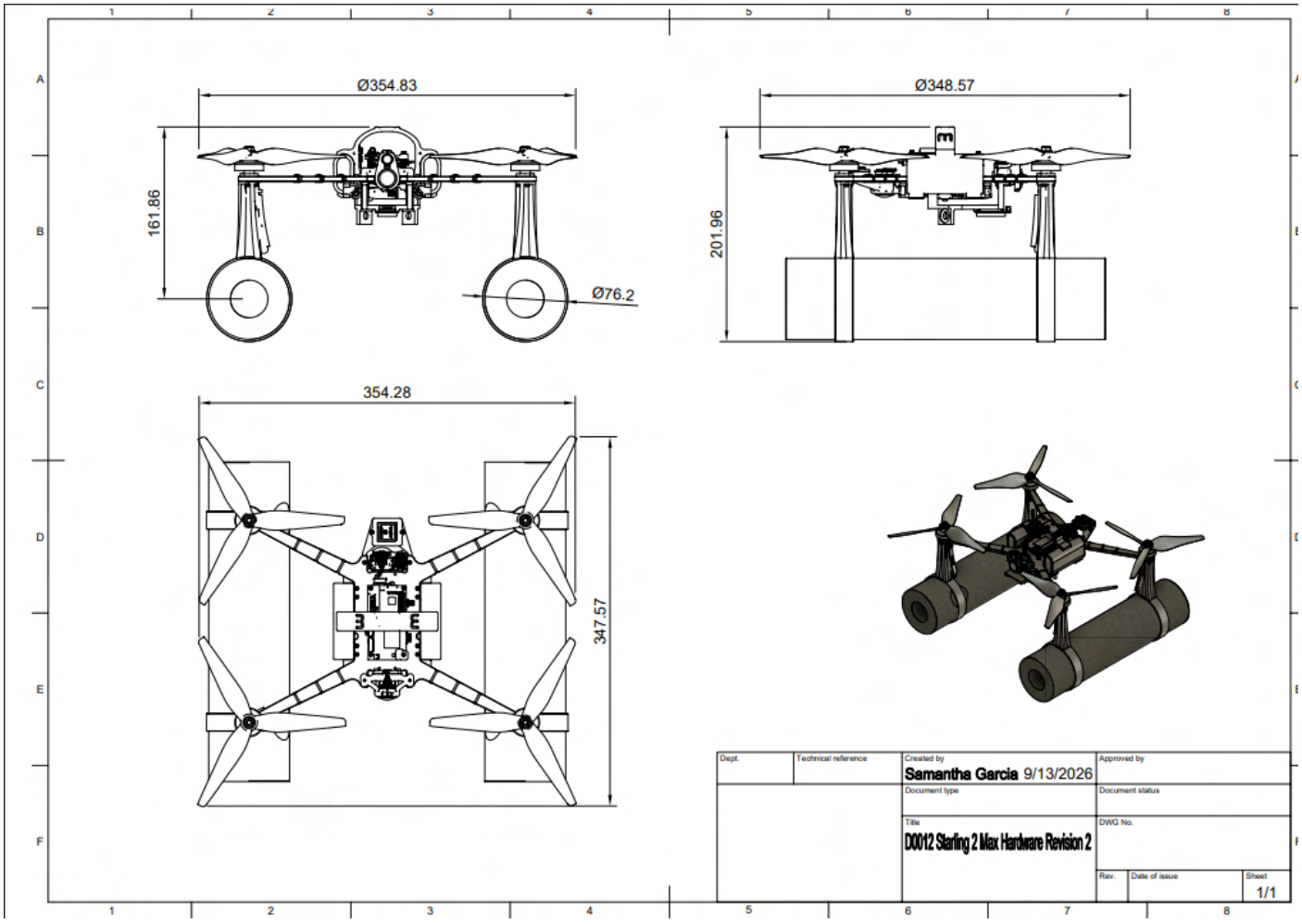


Fig. Q.1. Modified Starling 2 Max mechanical drawings.

APPENDIX R
STARLING 2 MAX WIRING

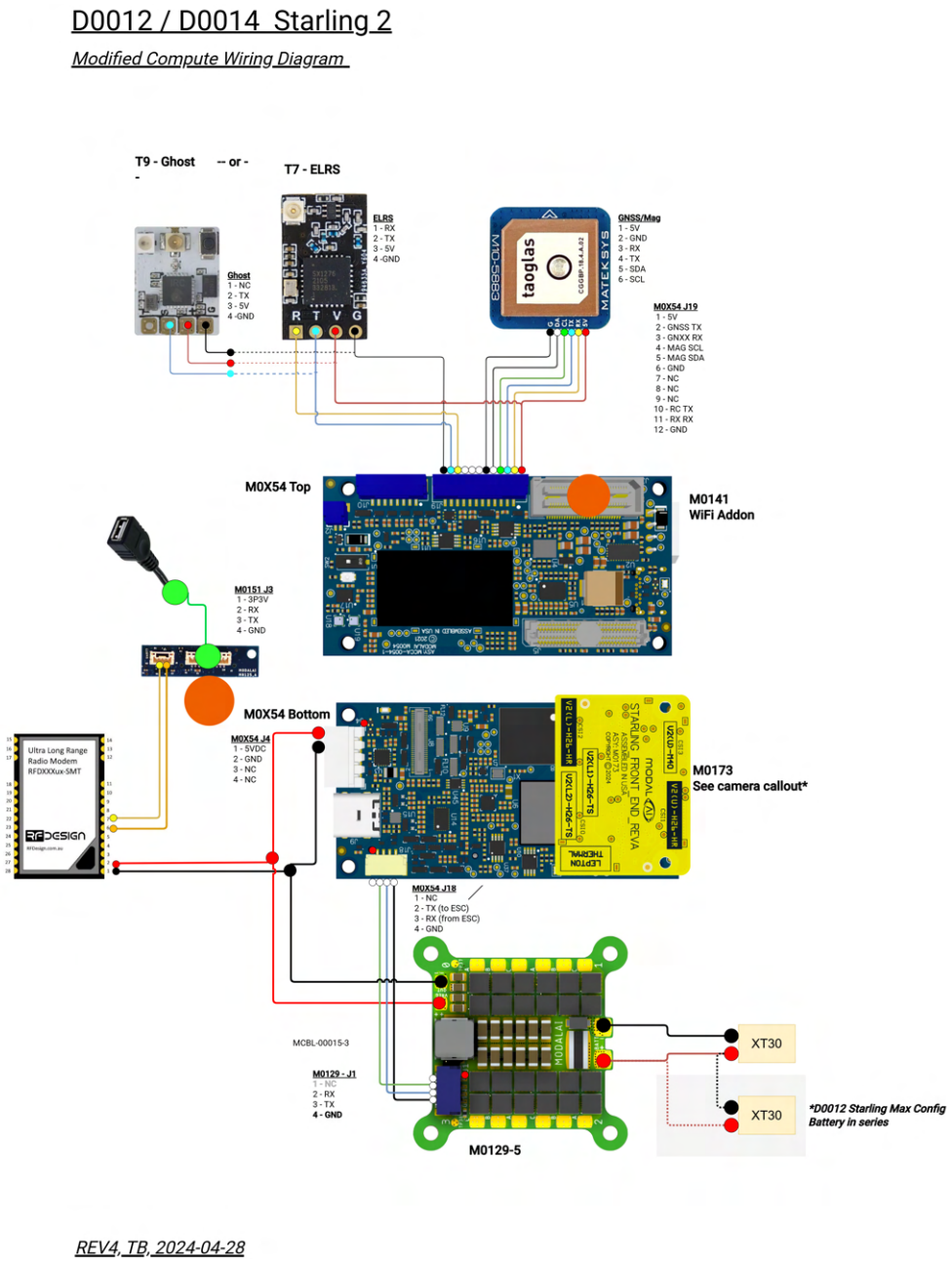


Fig. R.1. Starling 2 Max wiring diagram, modified from ModalAI [14].

APPENDIX S
STARLING 2 SENSOR WIRING

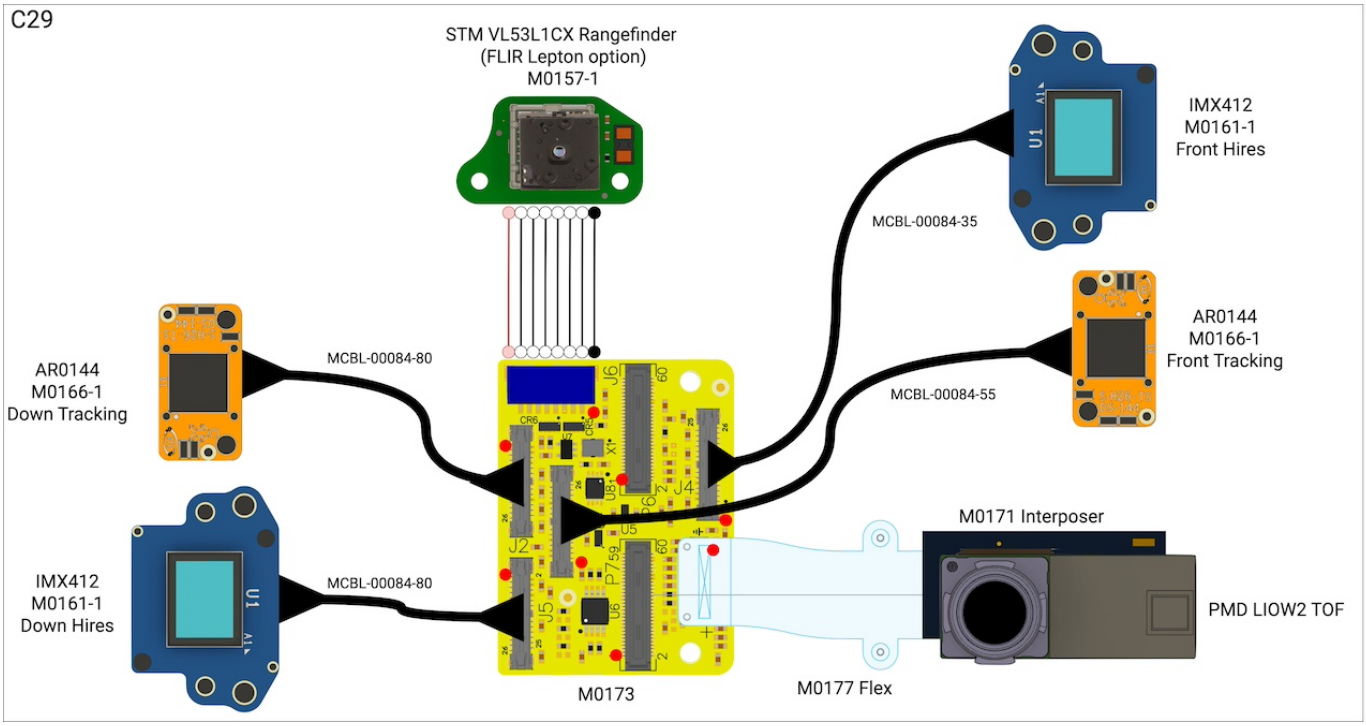


Fig. S.1. Starling 2 Max camera configuration. Reproduced from ModalAI [14].

APPENDIX T
STARLING 2 MAX PREFLIGHT CHECKLIST

Platform: Starling 2 Max with VOXL 2, PX4, and QGroundControl (SeaOwl configuration).

A. Before Leaving the Bench

- ☐ **Battery:** Fully charged; verify that the cells are balanced and that no two cells differ by more than 0.05 V.
- ☐ **Transmitter:** Charged with the correct aircraft profile selected.
- ☐ **Propellers:** Inspect for chips, cracks, and stress marks. Verify the correct rotation direction on each arm and confirm that the propeller nuts are properly secured.
- ☐ **Frame and arms:** Check for cracks, loose fasteners, and damaged motor mounts.
- ☐ **Motors:** Rotate each motor by hand and verify that it moves freely without rough or gritty bearings.
- ☐ **Camera and payload:** Confirm that all components are secure and that the camera lenses are clean. The `hires_front` camera is required by the beacon-detection pipeline.
- ☐ **Antennas:** Verify that the GPS/GNSS and RFD900ux antennas are attached and undamaged. Never power the radio without its antenna connected.
- ☐ **Operator station computer:** Confirm that the computer is charged, QGroundControl is available, and the correct communication link is configured, either Wi-Fi UDP or ELRS.

B. Regulatory and Site Checks

- ☐ **Airspace:** Confirm that the operating area is within authorized airspace. Obtain LAANC authorization when required.
- ☐ **Flight restrictions:** Check applicable Notices to Air Missions (NOTAMs) and Temporary Flight Restrictions (TFRs).
- ☐ **Weather:** Confirm that wind is within the aircraft's safe operating limits, no precipitation is present, visibility is at least 3 statute miles, and applicable cloud-clearance requirements are satisfied.
- ☐ **Registration and Remote ID:** Verify that the aircraft registration marking is visible and that Remote ID is broadcasting, unless operating under an applicable exception.

- ☐ **Operating area:** Confirm that the flight area is clear of uninvolved people, vehicles, and obstacles. Visual line of sight must be maintainable throughout the planned flight.
- ☐ **Crew briefing:** Review the mission, individual responsibilities, communications, abort criteria, and emergency procedures with all crew members.

C. Power-Up Sequence

- ☐ Place the aircraft on level ground and note its nose direction.
- ☐ Turn on the transmitter first. Verify that the throttle is low, the arming control on channel 8 is low or unlatched, and the kill switch on channel 5 is in its known safe position.
- ☐ Connect the flight battery and remain clear of the propellers during startup.
- ☐ Allow approximately 30–60 seconds for the VOXL 2 to boot. Confirm that the aircraft's status indicators are normal.

D. Communication and System Checks

- ☐ Confirm that QGroundControl connects through Wi-Fi UDP or the RFD900ux link. Verify continuous telemetry for at least 30 seconds without repeated link interruptions.
- ☐ If a system appears abnormal, inspect the VOXL services using `voxl-inspect-services`. Confirm that `voxl-px4` and `voxl-mavlink-server` are running.
- ☐ Confirm that QGroundControl displays no red pre-arm errors. Do not bypass pre-arm safety checks during field operations.
- ☐ **GPS/GNSS:** Confirm a three-dimensional position lock, at least 10 satellites, and HDOP below 1.5. Monitor for EKF2 GPS drift and height-instability warnings. Do not use Position mode if the estimated position moves while the aircraft is stationary.
- ☐ **Battery telemetry:** Confirm that the battery voltage shown in QGroundControl is correct and consistent with a fully charged pack.
- ☐ **Heading:** Confirm that the displayed heading agrees with the physical nose direction. Rotate the aircraft approximately 90° and verify that the displayed heading changes accordingly.
- ☐ **Flight modes:** Verify the flight mode shown in the QGroundControl toolbar. Move the channel 6 mode switch through each position and confirm that the displayed mode changes correctly.
- ☐ Return the mode switch to **Position** mode. Do not leave the switch in Mission mode before takeoff.
- ☐ **RC controls:** Verify in the QGroundControl Radio panel that all four control sticks move the correct indicators through their full expected ranges.
- ☐ **Kill switch:** Toggle channel 5 and verify that QGroundControl correctly indicates kill-switch engagement and disengagement.
- ☐ If an autonomous mission will be flown, verify that the mission is uploaded correctly, the first waypoint is reasonable, and the RTL altitude is above nearby obstacles.

E. Immediately Before Takeoff

- ☐ Install the propellers only after completing checks that require their removal. Confirm correct placement, rotation direction, and attachment.
- ☐ Place the aircraft on a level launch surface and establish a safe separation distance from all personnel.
- ☐ Confirm that QGroundControl has established the home position and that the "H" marker corresponds to the intended RTL location.
- ☐ Confirm that the RC-loss action, telemetry-loss behavior, low-battery actions, RTL altitude, and geofence settings are appropriate for the flight.
- ☐ Announce the intention to arm the aircraft to all nearby personnel.
- ☐ Arm the aircraft using the channel 8 control. Confirm that the motors idle normally without unusual vibration, noise, or warning tones.
- ☐ Perform a brief hover at approximately 1–2 m. Check for excessive drift, verify control response in all axes, and monitor QGroundControl for warnings.
- ☐ If abnormal behavior is observed, land and disarm immediately. Diagnose the problem while the aircraft is safely on the ground.

F. Abort Criteria

Do not launch, or land the aircraft as soon as safely practicable, if any of the following conditions occur:

- EKF or GPS warnings appear, or the estimated position visibly drifts.
- The telemetry link repeatedly disconnects.
- Battery voltage sags more than expected during hover.
- An unexpected flight-mode change or failsafe action occurs.
- The aircraft exhibits excessive vibration, unusual noise, or sluggish control response.
- Weather or site conditions exceed the established operating limits.
- Visual line of sight can no longer be maintained.

The aircraft is considered safe for handling only when it is disarmed and the flight battery is disconnected. The kill switch must not be treated as a substitute for disarming and disconnecting the aircraft.

APPENDIX U
STARLING 2 MAX MODIFICATION AND BUILD LOG

Samantha Garcia
Center for Connected Autonomy and Artificial Intelligence
College of Electrical Engineering & Computer Science
Florida Atlantic University
Boca Raton, FL, USA
scruz10@fau.edu

This appendix records the baseline configuration, controlled modifications, current as-built configuration, interface changes, quantitative impacts, software changes, and validation results for the ModalAI Starling 2 Max UAV. The source workbook contains 37 modification records through September 24, 2026. Historical status entries describe the state at the time of each record; the current-configuration table should be used when determining the latest as-built state.

A. Aircraft Baseline

TABLE U.1: Starling 2 Max baseline configuration.

Item	Baseline Value	Item	Baseline Value
Platform	ModalAI Starling 2 Max	VOXL / Companion Computer	VOXL2
Serial Number	M2500000JUU	Flight Controller / PX4	PX4
Registration / FAA ID	FA3WATHAK3	PX4 Firmware	1.14.0
Baseline Date	2025-10-02	VOXL Software	1.5.0
Baseline Flight Hours	0	ROS / ROS 2	ROS2 - foxy
Baseline Weight (g)	566	RC Receiver	915mhz ELRS
Baseline CG X (mm)	450	Telemetry / Data Link	WiFi: AlfaNetworks AWUS036EACS
Baseline CG Y (mm)	380	GPS / GNSS	UBlox M10
CG Reference / Origin	ModalAI Starling 2 Max Datasheet	Camera / Vision Sensors	ModalAI dual M0161, dual M0166
Primary Mission / Use	RobotX competition	Battery Configuration	Two 2S Li-ion 18650 in series with XT30 connector
Motors	2203.5 1500kv	Propellers	180mm Tri-Blade

Baseline note: Received development drone used by another team that was reflashed. Had custom GPS mount as noted in MOD-001. No crash history was reported prior to these logs

B. Modification History

TABLE U.2: Starling 2 Max modification history.

Mod ID	Date	Flight h	System	Change	Configuration and Objective	Status
MOD-001	2026-06-22	0	Sensors	Relocate; GPS/GNSS Mount	**New:** Raised GPS mount to above other components; **Objective:** Reduce interference from onboard electronics	**Passed**
MOD-002	2026-06-22	0	Communications	Add; RFD900x radio & mount	**New:** RFD900x radio installed on bottom of aircraft and connected to VOXL2 UART add-on board; **Objective:** Provide long-range MAVLink telemetry / QGroundControl communication; **Notes:** Did not maintain reliable connection	**Flight Tested**

Continued on next page

TABLE U.2 (continued)

Mod ID	Date	Flight h	System	Change	Configuration and Objective	Status
MOD-003	2026-06-22	0	Software	Software Change; Uploaded beacon detector script	**New:** Beacon detector script uploaded and started using shell script within a docker container; **Objective:** Beacon detector required for mission; **Notes:** Ran services but did not start beacon detector properly via system service	**Flight Tested**
MOD-004	2026-07-02	3.5	Software	Software Change; Added lawnmower flight pattern script	**New:** Added autonomous flight script to fly drone in lawnmower pattern; **Objective:** Test autonomous flight outside of QGC; **Notes:** Drone successfully flew in lawnmower pattern using shell script	**Passed**
MOD-005	2026-07-02	3.5	Communications	Replace; Switched back connection to WiFi Dongle	**New:** RFD900x radio removed and wifi dongle re-attached; **Objective:** RFD900x radio connection was unreliable; **Notes:** Wifi successfully reconnected & radio removed	**Passed**
MOD-006	2026-07-02	3.5	Communications	Remove; Removed rfd900x radio & mount	**New:** RFD900x radio removed and wifi dongle re-attached; **Objective:** RFD900x radio connection was unreliable; **Notes:** Wifi successfully reconnected & radio removed	**Passed**
MOD-007	2026-07-06	4.3	Software	Software Change; Added system service to program flight script to RC input	**New:** Lawnmower pattern script triggered by button press on RC controller; **Objective:** Need to start autonomous flight using RC controller; **Notes:** button successfully started autonomous mission	**Passed**
MOD-008	2026-07-08	5.4	Communications	Rewire; RFD900ux radio, mount, & wiring	**New:** Power harness added to pull power from main board supply and use UART for radio, RFD900ux installed and connected to VOXL2 UART add-on board; **Objective:** Provide long-range MAVLink telemetry / QGroundControl communication while still using WiFi; **Notes:** Successfully connected to radio & wifi simultaneously	**Passed**
MOD-009	2026-07-08	5.4	Software	Software Change; Updated beacon detector script with ground truth validation	**New:** Added multiple system services for each ground truth; **Objective:** Need to validate beacon detector script for each color; **Notes:** Successfully ran service for each beacon configuration, color classification reliable but blink detection status failing	**Flight Tested**
MOD-010	2026-07-22	9.4	Software	Software Change; Updated beacon detector script	**New:** Updated beacon detector to use different hue values and updated blink detection code; **Objective:** Unreliable blink detection; **Notes:** Successfully ran but still unreliable blink detection	**Flight Tested**
MOD-011	2026-07-27	10.4	Software	Software Change; Updated beacon detector script	**New:** Updated beacon detector with different color parameters; **Objective:** Unreliable blink detection; **Notes:** Successfully ran but still unreliable blink detection	**Flight Tested**
MOD-012	2026-08-13	10.6	Airframe	Replace; Landing gear	**New:** New landing gear using TPU to create 3/4" pvc fitting attached to drone, PVC landing equipment with pool noodles; **Objective:** Needed to make drone positively buoyant; **Notes:** Did not fly above a few meters altitude	**Failed**
MOD-013	2026-08-14	10.7	Airframe	Replace; Landing gear	**New:** Landing gear printed out of TPU with attachments directly to insert 3" pool noodles; **Objective:** Drone too heavy to fly reliably; **Notes:** Flew normally with new landing gear, positively buoyant in water	**Passed**
MOD-014	2026-08-17	10.9	Flight Control	Parameter Change; Failsafe parameters	**New:** Return to launch for loss of RC link after 1 second. Return to loss procedure rises to 65 feet and loiter for 10 seconds before landing, telemetry loss notification in QGC after 20 seconds, GPS/GNSS loss switches to altitude mode and then manual if no height estimate available, battery values set to 5, 10, 15% for emergency, critical, and warning. Geofence boundary changed to 1000 ft.; **Objective:** Need to have proper failsafe procedures in place; **Notes:** Need to perform more modifications due to some not being implemented in testing	**Failed**
MOD-015	2026-08-18	12.9	Airframe	Repair; Rear camera mount, GPS mount	**New:** 3d printed replacement parts using PETG filament; **Objective:** emergency landing caused parts to break; **Notes:** tested that cameras and radios work	**Bench Tested**
MOD-016	2026-08-18	12.9	Flight Control	Parameter Change; Waypoint navigation radius	**New:** 1.6 ft for NAV_ACC_RAD; **Objective:** the range was too high to fly waypont missions accurately; **Notes:** successful hover test and waypoint mission	**Passed**
MOD-017	2026-08-19	13.9	Software	Software Change; Autonomous flight triggered by button press	**New:** Autonomous flight to move in a triangular motion triggered by button press; **Objective:** Flight plan 3 for robotx competition; **Notes:** did not fly when button was pressed	**Failed**
MOD-018	2026-08-20	13.9	Software	Software Change; hover test triggered by button press	**New:** RC triggers bench test or flight test on channel 7; **Objective:** Flight plan 3 mission non-responsive with rc trigger; **Notes:** triggered response in bench test, hover flight on rc trigger	**Passed**
MOD-019	2026-08-20	14	Flight Control	Parameter Change; Input mode for manual controls	**New:** RC_COM_IN_MODE set to 0 (RC only); **Objective:** Need RTL response when RC input is lost; **Notes:** tested at bench, future testing in flight	**Passed**
MOD-020	2026-08-20	14	Software	Software Change; flight plan 3 triggered by rc command	**New:** RC triggers flight plan 3; **Objective:** Need to test flight plan 3 autonomously; **Notes:** did not fly as expected	**Failed**
MOD-021	2026-08-21	14.1	Software	Parameter Change; change speed for movement and turning	**New:** MPC_Z_VEL_MAX_UP = 3; MPC_XY_VEL_MAX = 8; MPC_Z_VEL_MAX_DN = 2; MPC_YAWRAUTO_MAX = 90; **Objective:** Movement was jerky for flight plan 3 testing; **Notes:** flew normally and did pirouettes in waypoint mission	**Passed**

Continued on next page

TABLE U.2 (continued)

Mod ID	Date	Flight h	System	Change	Configuration and Objective	Status
MOD-022	2026-08-21	14.5	Software	Software Change; disable system service for flight plan 3	**New:** seabird-mission service disabled; **Objective:** Waypoint mission created to complete pirouette mission autonomously - service no longer needed; **Notes:** verified service was no longer active after reboot	**Passed**
MOD-023	2026-09-02	15	Software	Software Change; Update configs for flights and add burst beacon detector	**New:** beacon-detector services starts burst_beacon_detector.py with new configs; **Objective:** Latency in detections caused incorrect beacon detections; **Notes:** Need to do more validation	**Flight Tested**
MOD-024	2026-09-02	15	Communications	Other; RFD900ux radio power	**New:** Change power to 15 dBm ground station and 5dBm on drone; **Objective:** interference with WiFi causing unreliable connection; **Notes:** Need to test for reliable connection in flight	**Ground Tested**
MOD-025	2026-09-03	15.5	VOXL2 / Compute	Parameter Change; XRCE_DDS Domain ID Configuration	**New:** 8; **Objective:** unable to see GPS topics in ROS2; **Notes:** able to see GPS topics in ROS2 when voxl-mps-ros2 node active	**Passed**
MOD-026	2026-09-10	15.5	Software	Software Change; beacon detector program by color	**New:** Added beacon detector that uses 4 color beacon model and only checks beacon status for blue beacons; **Objective:** blink detection algorithm too CPU intensive & does not detect consistently; **Notes:** Still debugging with previous config files that stream into this script	**Not Tested**
MOD-027	2026-09-10	16	Software	Software Change; beacon detector program	**New:** Use GPS to determine pose and height, log gps output in print statements; **Objective:** VIO is not used for outdoor flight; **Notes:** Detected beacons but GPS topics were not collecting information in script	**Failed**
MOD-028	2026-09-11	16.1	Software	Software Change; beacon detector program by color	**New:** if px4_sensor_gps is message type, use BEST_EFFORT Reliability Policy and KEEP_LAST History Policy, depth = 1; **Objective:** GPS topics were not visible in script; **Notes:** GPS was recorded in single flight test, but could not continue testing due to weather	**Flight Tested**
MOD-029	2026-09-11	16.1	Airframe	Replace; GPS Mount	–	**–**
MOD-030	2026-09-17	16.1	Communications	Remove; RFD900ux	–	**–**
MOD-031	2026-09-17	16.1	Airframe	Remove; Radio mount	–	**–**
MOD-032	2026-09-17	16.1	Communications	Replace; RX Module	**New:** 2.4 GHz ELRS link; **Objective:** Singapore regulations do not allow 915MHz	**Bench Tested**
MOD-033	2026-09-17	16.1	Communications	Replace; TX Module	**New:** 2.4 GHz Nano external TX module; **Objective:** Singapore regulations do not allow 915MHz	**Bench Tested**
MOD-034	2026-09-23	16.1	Communications	Software Change; RFD900ux system service	**New:** rfd900-mavlink service disabled; **Objective:** No longer using rfd900ux radios so service no longer needed; **Notes:** system service not started on reboot	**Passed**
MOD-035	2026-09-22	16.1	Airframe	Add; Payload mounting gear	**New:** Payload mounting attachment; **Objective:** Needed to be able to mount robotic gripper for payload delivery	**Passed**
MOD-036	2026-09-24	16.1	Airframe	Replace; Landing gear	**New:** Inflatable tubes used for buoyant landing gear; **Objective:** Needed to reduce mass for payload attachment; **Notes:** verified buoyancy - plan for flight testing	**Bench Tested**
MOD-037	2026-09-24	16.1	Communications	Add; Telemetry radio	**New:** ELRS module configured as telemetry radio; **Objective:** Needed telemetry radio compliant with Singapore regulations	**Pending**

C. Current As-Built Configuration

TABLE U.3: Current Starling 2 Max as-built configuration.

System	Component	Manufacturer / Model	Interface / Power	Status	Notes
Navigation	GPS / GNSS	u-blox; M10	**Port:** J19 - GPS UART + Mag I2C; **Endpoint:** Existing GPS interface; **Power:** Starling power	Installed	**Introduced:** MOD-001; **Notes:** GPS relocated/remounted; orientation unchanged
Communications	Wi-Fi Dongle	TP-Link; Archer T2U Plus AC600	**Port:** Starling add-on board Wi-Fi interface; **Endpoint:** wlan0; **Power:** Add-on board 5 V	Installed	**Introduced:** MOD-005; **Notes:** Wi-Fi connection restored
Communications	Telemetry Radio	RFDesign; RFD900ux-US	**Port:** UART add-on board; **Endpoint:** /dev/ttyHS1; **Power:** Starling 2 power module	Removed	**Introduced:** MOD-008; **Notes:** Custom harness; UART signal from add-on board, separate power from power module
Software	MAVLink / RFD Bridge	Custom; rfd900_mavlink_bridge.py	**Port:** Internal software; **Endpoint:** MAVLink/UDP + / dev/ttyHS1; **Power:** N/A	Disabled	**Introduced:** MOD-002; **Notes:** Bidirectional RFD900ux MAVLink bridge
Software	Beacon Detector	Custom; beacon_detector_config.py	**Port:** Internal / ROS 2; **Endpoint:** ROS 2 / VOXL / MPA; **Power:** N/A	Installed	**Introduced:** MOD-011; **Notes:** Current beacon detector version
Software	MPA-to-ROS2 Service	Custom / ModalAI package; voxl-mpa-ros2.service	**Port:** Internal; **Endpoint:** ROS 2 / VOXL / MPA pipes; **Power:** N/A	Installed	**Introduced:** MOD-003; **Notes:** Automatically launches MPA-to-ROS2 bridge
Airframe	Landing gear	Custom; Current landing gear	**Port:** N/A; **Endpoint:** N/A; **Power:** N/A	Installed	**Introduced:** MOD-013; **Notes:** Current final landing gear configuration that is positively buoyant

Continued on next page

TABLE U.3 (continued)

System	Component	Manufacturer / Model	Interface / Power	Status	Notes
Airframe	Rear downward camera mount	ModalAI design; 951A	**Port:** N/A; **Endpoint:** N/A; **Power:** N/A	Installed	**Introduced:** MOD-015; **Notes:** Replacement due to emergency landing
Communications	RC Control	RadioMaster; RadioMaster RP1 V2, 2.4 GHz ExpressLRS, CRSF UART	**Port:** VOXL2 RC UART / CRSF; **Endpoint:** PX4 input_rc / rc_channels topics; **Power:** Starling power	Installed	**Introduced:** MOD-032; **Notes:** Need comms that are legal in Singapore

D. VOXL 2 Ports and Wiring

TABLE U.4: VOXL 2 port and wiring changes recorded in the modification log.

Conn. ID	Mod ID	Source to Destination	Interface	Power / Cable	Active	Notes
CONN-001	MOD-002	**From:** ModalAI M0151 USB3/ UART Add-On; **Port:** J3 - UART; **To:** RFD900x; **Port:** UART / Serial	**Type:** UART / MAVLink; **Endpoint:** /dev/ttyHS1; **Rate:** 57600 baud	**Voltage:** N/A - signal connection; **Source:** N/A - see separate power connection; **Cable:** Custom UART cable	No	RFD900x telemetry link used for bidirectional MAVLink/ QGroundControl communication
CONN-002	MOD-002	**From:** ModalAI M0151 USB3/ UART Add-On; **Port:** USB / Wi-Fi peripheral power; **To:** RFD900x; **Port:** Power input	**Type:** Power; **Endpoint:** N/A; **Rate:** N/A	**Voltage:** 5 V; **Source:** M0151 USB peripheral power; **Cable:** Modified/custom power cable	No	RFD900x powered from M0151 USB power feed normally used by Wi-Fi adapter
CONN-003	MOD-005	**From:** ModalAI M0151 USB3/ UART Add-On; **Port:** USB / Wi-Fi peripheral power; **To:** Wifi Dongle; **Port:** USB	**Type:** USB; **Endpoint:** wlan0; **Rate:** N/A	**Voltage:** 5 V; **Source:** USB power from ModalAI add-on board; **Cable:** USB adapter / dongle connection	Yes	Verified wlan interface, IP assignment, SSH/network access, and coexistence with RFD900 telemetry
CONN-004	MOD-008	**From:** ModalAI UART Add-On Board; **Port:** J3 - UART; **To:** RFD900ux; **Port:** UART / Serial	**Type:** UART / MAVLink; **Endpoint:** /dev/ttyHS1; **Rate:** 57600 baud	**Voltage:** N/A - signal connection; **Source:** N/A - see separate power connection; **Cable:** Custom UART cable	Yes	Provides bidirectional MAVLink telemetry between VOXL2 and RFD900ux
CONN-005	MOD-008	**From:** ModalAI UART Add-On Board; **Port:** 5 V power output / modified power tap; **To:** RFD900ux; **Port:** Power input	**Type:** Power; **Endpoint:** N/A; **Rate:** N/A	**Voltage:** 5 V; **Source:** Starling 2 Max power module; **Cable:** Custom modified power/UART harness	Yes	RFD900ux powered directly from aircraft power module rather than UART/USB board

E. Power Impact

The workbook assigns 10 W of accessory power and reserves a 20% margin. Its enabled-load calculation reports 0.30 W typical, 3.25 W maximum, and 4.75 W of remaining maximum margin.

TABLE U.5: Recorded accessory power impact.

Component	Mod ID	Voltage	Typical W	Max W	Power Source	Enabled	Notes
RFD900x Telemetry Radio	MOD-002	5	0.3	5	M0151 Wi-Fi power feed	No	Continuous during telemetry use; Initial telemetry configuration; Wi-Fi unavailable while this configuration was used
Wi-Fi Dongle	MOD-005	5	–	2.5	ModalAI add-on board Wi-Fi power connection	Yes	Continuous when Wi-Fi enabled; load varies with traffic; Wi-Fi restored after radio wiring change
RFD900ux Telemetry Radio	MOD-008	5	0.3	5	Starling 2 Max power module	No	Continuous during telemetry use; Current telemetry radio; signal from UART board, power from aircraft power module
RFD900ux Telemetry Radio	MOD-024	5	0.3	0.75	M0151 Wi-Fi power feed	Yes	Continuous during telemetry use; Current telemetry radio; signal from UART board, power from aircraft power module

F. Weight and Center-of-Gravity Impact

The workbook reports a baseline mass of 566 g and a calculated current mass of 672.96 g. The center-of-gravity moment calculation is not complete because the coordinate reference and several component positions remain undefined.

TABLE U.6: Recorded modification mass and center-of-gravity impact.

Mod ID	Component	Action	Δm (g)	Mount Location	Basis	Permanent	Notes
MOD-001	GPS/GNSS Mount	Added	4.2	Original GPS mount position, 150 mm higher	Estimated	Yes	New custom GPS mount
MOD-002	RFD900x radio & mount	Added	77.4	Bottom of aircraft	Estimated	No	–
MOD-006	RFD900x radio & mount	Removed	-77.4	Bottom of aircraft	Estimated	Yes	–

Continued on next page

TABLE U.6 (continued)

Mod ID	Component	Action	Δm (g)	Mount Location	Basis	Permanent	Notes
MOD-008	RFD900ux radio, mount & wiring	Added	37.4	Bottom of aircraft	Estimated	Yes	–
MOD-012	Landing gear	Added	471	Bottom of aircraft	Estimated	No	–
MOD-013	Landing gear	Removed	-371	Bottom of aircraft	Estimated	Yes	–
MOD-030	RFD900ux radio, mount & wiring	Removed	-37.4	Bottom of aircraft	Estimated	Yes	–
MOD-032	BetaFPV ELRS Nano Receiver, 915 MHz & antenna	Removed	-1.64	Top of aircraft beside VOXL2	Estimated	Yes	–
MOD-032	RadioMaster RP1 V2 Receiver, 2.4GHz & antenna	Added	2.2	Top of aircraft beside VOXL2	Estimated	Yes	for RC link
MOD-033	RadioMaster RP1 V2 Receiver, 2.4GHz & antenna	Added	2.2	Top of aircraft beside VOXL2	Estimated	Yes	second unit for telemetry link

G. Software, PX4, ROS 2, and MAVLink Changes

TABLE U.7: Software and configuration change log.

Config ID	Date	Mod ID	Device / Service	Parameter / File	Previous	New	Validated
CFG-001	2026-06-22	MOD-002	RFD900x MAVLink Bridge	rfd900_mavlink_bridge.py	–	/dev/ttyHS1, 57600 baud, UDP MAVLink bridge	Yes
CFG-002	2026-06-22	MOD-003	Beacon Detector	beacon_detector_config.py	–	–	Yes
CFG-003	2026-06-22	MOD-003	Link voxl mpa to ROS2	/etc/systemd/system/ voxl-mpa-ros2.service	Manual launch	Automatic systemd-managed startup	Yes
CFG-004	2026-06-22	MOD-003	Beacon Detector Service	/etc/systemd/system/ beacon-detector.service	Manual launch	Automatic systemd-managed startup	Yes
CFG-005	2026-06-22	MOD-003	Start Docker Container for Custom scripts	run_seabird.sh	Start via user profile and docker up	Shell script to automatically start docker container	Yes
CFG-006	2026-06-22	MOD-003	Generate DDS profile for ROS2 link to docker container	fastdds_profile.xml	No dds profile generated	Generate dds profile upon startup for beacon detector script and docker container	Yes
CFG-007	2026-07-02	MOD-004	Autonomous Lawnmower Flight	sweep_lawnmower.py	–	Start autnomous flight using shell script	Yes
CFG-008	2026-07-06	MOD-007	Autonomous Flight Service	sweep_lawnmower.py	Manual launch	Automatic systemd-managed startup	Yes
CFG-009	2026-07-08	MOD-009	Beacon Detector Service with Ground Truths	beacon_detector_config.py	Single configuration	systemd-managed startup for each ground truth configuration	Yes
CFG-010	2026-07-22	MOD-010	Beacon Detector Script update	beacon_detector_config.py	Different hue values and blink detection	Updated hue values and blink detection	Yes
CFG-011	2026-07-27	MOD-011	Beacon Detector Script update	beacon_detector_config.py	Different color parameters for blink detection	Used color parameters to determine blink status	Yes
CFG-012	2026-08-17	MOD-014	RC Link Failsafe	COM_RC_LOSS_T, NAV_RCL_ACT, RTL parameters, COM_DL_ LOSS_T, position-loss parameters, BAT_LOW_ THR, BAT_CRIT_THR, BAT_EMERGEN_THR, GF_MAX_HOR_DIST	Previous/default failsafe configuration	RC loss to RTL after 1 s; RTL climbs to 65 ft, loiters 10 s, then lands; telemetry loss indication after 20 s; position loss to Altitude/ Manual fallback; battery thresholds 15% warning, 10% critical, 5% emergency; max horizontal geofence 1000 ft	Yes
CFG-013	2026-08-18	MOD-016	Waypoint navigation radius	NAV_ACC_RAD	32.8 ft	1.6 ft	Yes
CFG-014	2026-08-19	MOD-017	Flight plan 3 autonomous mission	element3_safety_check.py	lawnmower pattern triggered by rc when service is enabled	element3 safety check triggered by rc when service is enabled	Yes
CFG-015	2026-08-20	MOD-018	Added hover test and bench test for rc trigger	hover_test.py	no hover or bench test for rc trigger service	hover_test.py triggered by seabird-mission service	Yes
CFG-016	2026-08-20	MOD-019	Modified input mode	COM_RC_IN_MODE	3 - RC and Joystick with fallback	0 - RC Transmitter only	Yes
CFG-017	2026-08-20	MOD-020	Switched to flight plan 3 for autonomous mission	start_flight_3.sh	started hover_test.py	start element3_safety_ check.py	Yes
CFG-018	2026-08-21	MOD-021	parameters relevant to pirouette script	MPC_Z_VEL_MAX_UP	10	3	Yes
CFG-019	2026-08-21	MOD-021	parameters relevant to pirouette script	MPC_XY_VEL_MAX	4.997	8	Yes
CFG-020	2026-08-21	MOD-021	parameters relevant to pirouette script	MPC_Z_VEL_MAX_DN	4	2	Yes
CFG-021	2026-08-21	MOD-021	parameters relevant to pirouette script	MPC_YAWRAUTO_MAX	45	90	Yes
CFG-022	2026-08-22	MOD-022	Disabled rc trigger service	seabird-mission.service	enabled	disabled	Yes

Continued on next page

TABLE U.7 (continued)

Config ID	Date	Mod ID	Device / Service	Parameter / File	Previous	New	Validated
CFG-023	2026-09-02	MOD-023	Beacon Detector Script	burst_beacon_detector.py	Used beacon_detector_ config.py	Use burst_beacon_detector.py to detect set of frames, calling functions from beacon_ detector_config.py	Yes
CFG-024	2026-09-02	MOD-023	Service update	burst_beacon_detector.py	Used beacon_detector_ config.py	Run burst_beacon_detector.py instead of beacon_detector_ config.py	Yes
CFG-025	2026-09-03	MOD-025	PX$ XRCE-DDS Client	XRCE_DDS_DOM_ID	0	8	Yes
CFG-026	2026-09-10	MOD-026	Beacon Detector Script	staged_beacon_detector.py	Used single model for beacon detection and single model for beacon top, then color detection & blink status pipeline	Used 4 color model for beacon detection, assume blinking for red & green, use original pipeline for blue or unknown	No
CFG-027	2026-09-10	MOD-027	Beacon Detector Script update	burst_beacon_detector.py, beacon_detector_config.py, beacon_detector.py, beacon_camera.py	Used VIO to determine drone pose & height	Use GPS to determine drone pose & height	Yes
CFG-028	2026-09-11	MOD-028	Beacon Detector Script	burst_beacon_detector.py, beacon_detector_config.py, beacon_detector.py, beacon_camera.py	Change	QoSProfile used RELIABLE Reliability Policy and TRANSIENT_LOCAL Durability Policy, depth = 1	Yes

H. Post-Modification Validation

TABLE U.8: Post-modification test and validation log.

Test ID	Date	Mod ID	Level	System	Test	Actual Result	Status	Notes
TEST-001	2026-06-22	MOD-001	Ground	Navigation	GPS lock and compass operation	Expected	**Pass**	Able to lock on 27 GPS satellites
TEST-002	2026-06-22	MOD-001	Flight	Navigation	Short hover / flight test	Expected	**Pass**	Flew with GPS logged normally
TEST-003	2026-06-22	MOD-002	Ground	Communications	MAVLink / QGC telemetry test	Parameters not uploading with RFD connection in QGC	**Fail**	Did not upload parameters in QGC or waypoint mission
TEST-004	2026-06-22	MOD-002	Flight	Communications	In-flight telemetry test	Maintained QGC link but did not upload mission	**Pass**	Maintained QGC connection but did not have waypoint mission uploaded
TEST-005	2026-07-02	MOD-005	Ground	Networking	Wi-Fi interface and SSH test	Connected to normally to wi-fi dongle, did not upload all parameters & waypoint mission reliably	**Fail**	Able to connect via wi-fi through ssh and QGC, but parameters did not upload in QGC and waypoint mission did not load
TEST-006	2026-07-08	MOD-008	Ground	Communications	UART/MAVLink test	Able to communicate both ways with radio and upload parameters & missions in QGC	**Pass**	Able to connect to QGC with RFD900ux radio connection and Wi-Fi as backup
TEST-007	2026-07-08	MOD-008	Flight	Communications	Telemetry flight test	Expected	**Pass**	QGC remained connected throughout several waypoint mission flights
TEST-008	2026-06-22	MOD-003	Ground	Software/Vission	Detection test	Detected beacon and color correctly but mis-classified blink status	**Fail**	–
TEST-009	2026-06-22	MOD-003	Flight	Software/Vission	Real-world detection test	Beacon detector script did not properly subscribe to image topics and detect beacon	**Fail**	–
TEST-010	2026-07-02	MOD-004	Flight	Flight Control	Autonomous pattern flight	Expected	**Pass**	–
TEST-011	2026-07-06	MOD-007	Flight	Software / RC	RC trigger test	Expected	**Pass**	–
TEST-012	2026-07-13	MOD-012	Ground	Airframe	Buoyancy inspection	Expected	**Pass**	–
TEST-013	2026-07-13	MOD-012	Flight	Airframe	Takeoff/landing test	Drone took off but struggled to lift and maintain heading	**Fail**	–
TEST-014	2026-07-14	MOD-013	Ground	Airframe	Buoyancy inspection	Expected	**Pass**	–
TEST-015	2026-07-14	MOD-013	Flight	Airframe	Takeoff/landing test	Expected	**Pass**	–

I. Supporting Evidence

TABLE U.9: Supporting evidence identified in the modification workbook.

Description	Location	Repository	Created By
Flight testing documentation	https://fau.sharepoint.com/:w:/r/sites/CAAIDroneClubSeabirdTeam/_layouts/15/Doc.aspx?action=edit&sourcedoc=%7B742ddcb0-480a-4490-9c85-a3886ca52901%7D&wdExp=TEAMS-TREATMENT&web=1	sharepoint file	Samantha Garcia
Flight testing videos	https://www.youtube.com/playlist?list=PLfD8SnXv125c	youtube	Samantha Garcia & Jaden Cirius

APPENDIX V
FLIGHT LOG

Date	Model	Reg #	Location	Flight Purpose	Duration	Start Time	Weather	Remarks
06/22/2026	ModalAI Starling2	FA3WATHAK3	8788 Senator Joe Abruzzo Avenue, Boynton Beach, FL	Initial test flight with ModalAI drone. Test autonomous mission in QGroundControl. Test blinking beacon script in each orientation	3.0	11:00 AM	Sunny, clear skies	Camera was not correctly publishing to blinking beacon code. Problem was troubleshot and working by the time we left. Could not get mission to upload in QGC using RFD900x radio; parameters took longer than expected to load over radio.
06/23/2026	ModalAI Starling2	FA3WATHAK3	Sam's house, Loxahatchee, FL	Test beacon detection code in each orientation	1.0	12:00 PM	Sunny, clear skies	Switched to WiFi AP for flight and could not get camera topics to publish. Modified services to check for topics before starting flight. Short test flights confirmed publishing.
06/24/2026	ModalAI Starling2	FA3WATHAK3	Sam's house, Loxahatchee, FL	Get more beacon data for fine-tuning training	1.0	12:00 PM	Sunny, clear skies	Successfully collected images and created labels from drone footage for beacon and beacon tops.
06/30/2026	ModalAI Starling2	FA3WATHAK3	Bob's house, Boca Raton, FL	Debug autonomous waypoint mission problems, collect additional beacon data	2.0	2:00 PM	Sunny, clear skies	Unable to get comms working consistently, so did not fly. Gathered beacon data by holding drone with onboard camera.
07/01/2026	ModalAI Starling2	FA3WATHAK3	Sam's house, Loxahatchee, FL	Test waypoint autonomous flight after calibration; test new blinking beacon code with ground truth	1.5	6:30 PM	Sunny, clear skies	Could not determine why waypoints were not working, but flew manually with consistent comms. Tested beacon detector in each configuration and obtained results for evaluation.
07/02/2026	ModalAI Starling2	FA3WATHAK3	Sam's house, Loxahatchee, FL	Test waypoint autonomous flight, test .sh script lawnmower pattern; get flight logs	0.8	7:00 PM	Clear skies	Recalibrated drone outside as suggested by online forums. Drone armed but did not follow waypoints. Shell script gave an error.
07/06/2026	ModalAI Starling2	FA3WATHAK3	Sam's house, Loxahatchee, FL	Test autonomous flight script based on button press	0.3	7:30 PM	Humid, clear skies	Drone armed and took off, started lawnmower pattern, but pattern was inconsistent. Lost offboard signal and returned home. Turns were aggressive; will slow them down for more reliable flight.
07/08/2026	Custom ARK	FA3RPRMHXP	Sam's house, Loxahatchee, FL	Initial test flight	0.1	9:30 AM	Sunny, clear skies	Drone propellers did not rotate at the same time. Drone did not take off and crashed. Need to replace motor mounts with COTS mounts and replace one ESC holder and landing gear.

Date	Model	Reg #	Location	Flight Purpose	Duration	Start Time	Weather	Remarks
07/08/2026	ModalAI Starling2	FA3WATHAK3	Sam's house, Loxahatchee, FL	Test waypoint mission with RFD900ux radio, test new blinking beacon code	1.0	3:00 PM	Sunny, clear skies	RFD900ux connected with WiFi also connected. Drone successfully executed waypoint mission twice. Blinking beacon code did not ID light colors correctly; will use unknown-beacon images to debug.
07/22/2026	ModalAI Starling2	FA3WATHAK3	Sam's house, Loxahatchee, FL	Tested beacon detector code and waypoint missions	4.0	12:00 PM	Sunny, clear skies	GPS did not detect enough satellites initially and vehicle would not arm. After securing wires and GPS mount, GPS worked, but waypoint missions did not. Did manual flights to detect beacon and test each config.
07/23/2026	ModalAI Starling2	FA3WATHAK3	FAU campus, EE building	Tested beacon detector code and waypoint missions	2.0	2:00 PM	Sunny, clear skies	Modified geofence and failsafe so drone did not return home above expected altitude. Successfully completed waypoint mission and ran beacon detector in each configuration. Color detection less accurate, possibly due to sun direction.
07/27/2026	ModalAI Starling2	FA3WATHAK3	Sam's house, Loxahatchee, FL	Tested beacon detector code and waypoint mission with code updates for color parameters	1.0	11:30 AM	Sunny, clear skies	Flew waypoint mission reliably for each of the 6 beacon configurations. Color detection was reliable, but blink detection had many false negatives.
08/13/2026	ModalAI Starling2	FA3WATHAK3	Sam's house, Loxahatchee, FL	Tested flight capability with new landing gear	0.20 * 2:15 PM	Sunny, clear skies		
08/14/2026	ModalAI Starling2	FA3WATHAK3	Sam's house, Loxahatchee, FL	Tested new flight capability with modified landing gear	0.10	4:00 PM	Sunny, clear skies	When in position flight mode, drone flew normally with landing gear attached. Able to attempt flight pattern and pirouettes outlined in proof of readiness for robotx.
08/17/2026	ModalAI Starling2	FA3WATHAK3	Sam's house, Loxahatchee, FL	failsafe testing and flight plan missions	2.00	2:00 PM	Sunny, clear skies	Showed kill switch and RTL works properly. Did not trigger appropriate failsafes for lost telemetry or loss of RC link. Attempted manual mission and had drift in altitude mode causing a crash. Replacement for rear downward camera mount, GPS mount, and propellers
08/18/2026	ModalAI Starling2	FA3WATHAK3	Sam's house, Loxahatchee, FL	Test flight after repairs and waypoint mission parameter update	1.00	5:00 PM	Sunny, clear skies	Replaced mount for downward facing camera and GPS mast. Recalibrated due to strong magnetic interference. Successfully armed and flew waypoint mission

Date	Model	Reg #	Location	Flight Purpose	Duration	Start Time	Weather	Remarks
08/19/2026	ModalAI Starling2	FA3WATHAK3	Sam's house, Loxahatchee, FL	Waypoint mission flight for robotx competition	1.00	1:30 PM	Sunny, clear skies	Completed first 2 flight tests. Third had waypoints that would not load properly, so attempted to set up autonomous mission using python script.
08/20/2026	ModalAI Starling2	FA3WATHAK3	Sam's house, Loxahatchee, FL	Hover test for autonomous script started by rc controller	0.40	3:00 PM	Sunny, clear skies	Successful test hover from channel 7 trigger - verified that RC can manually take over.
08/21/2026	ModalAI Starling2	FA3WATHAK3	Sam's house, Loxahatchee, FL	test flight plan 3 script	0.10	2:20 PM	Sunny, clear skies	Ran pirouettes script and first only hovered, then spun almost immediately without completing full waypoint paths; modifying parameters to test again.
08/21/2026	ModalAI Starling2	FA3WATHAK3	Sam's house, Loxahatchee, FL	Test waypoint pirouettes	0.50	5:30 PM	Sunny, clear skies	Successfully completed pirouettes using QGC waypoint mission - headings will be updated to match direction of drone for fp3.
08/24/2026	ModalAI Starling2	FA3WATHAK3	Sam's house, Loxahatchee, FL	Complete flight plan 3 and manual flights for POR robotx	1.00	11:30 AM	Sunny, clear skies	Successfully completed fp3 and most fail-safe testing before battery needed to be recharged.
08/24/2026	ModalAI Starling2	FA3WATHAK3	Sam's house, Loxahatchee, FL	Re-do for autonomous flight plan 1, 2, and manual flight plan 3. Test RC link loss	1.00	3:00 PM	Sunny, clear skies	Successfully completed manual and autonomous missions.
08/28/2026	ModalAI Starling2	FA3WATHAK3	Sam's house, Loxahatchee, FL	Test autonomous flight with geofence violations	0.50	10:00 AM	Sunny, clear skies	Failed testing; for horizontal GF, it flew past fence. For vertical, it did not take off.
08/28/2026	ModalAI Starling2	FA3WATHAK3	Sam's house, Loxahatchee, FL	Test autonomous flight with geofence violations	0.50	12:30 PM	Sunny, clear skies	Would not take off for missions that violated geofence or height restrictions in any waypoints.
09/01/2026	ModalAI Starling2	FA3WATHAK3	Sam's house, Loxahatchee, FL	Record vidoes of beacon for model training	0.50	12:00 PM	Sunny, clear skies	Recorded video of beacon in each configuration - blinking beacons recorded twice since video was zoomed in for first set.
09/02/2026	ModalAI Starling2	FA3WATHAK3	Sam's house, Loxahatchee, FL	Test beacon detector with burst beacon detector on drone	0.50	12:30 PM	Sunny, cloudy skies	Only ran first 2 beacon detectors because it was performing poorly overall.
09/03/2026	ModalAI Starling2	FA3WATHAK3	Sam's house, Loxahatchee, FL	Test beacon detector with simpler waypoint mission and new model	0.50	8:45 AM	Sunny, clear skies	beacons were detected better with new model, however did not accurately get beacon tops and colors. Images very overexposed and gps coordinate topics not visible. Next steps to investigate gps topic issue, improve color detection, and validate gps ground truth

Date	Model	Reg #	Location	Flight Purpose	Duration	Start Time	Weather	Remarks
09/04/2026	ModalAI Starling2	FA3WATHAK3	Sam's house, Loxahatchee, FL	Test GPS topic subscription and beacon detector. Gather new beacon data	0.50	9:00 AM	Sunny, clear skies	Able to gather beacon data, however did not get gps topic. Needed to modify Dockerfile for script to include px4_msgs.
09/09/2026	ModalAI Starling2	FA3WATHAK3	Sam's house, Loxahatchee, FL	Test GPS topic and beacon detector - gather beacon data	0.50	1:45 PM	Sunny, cloudy skies	Recalibrated with new GPS mount from factory and TOF sensor - difficulty connecting to wifi at first but then became more reliable after adjusting wifi antenna. Unreliable color detections but beacon was detected. Code was using VIO to get height, so never went to gps downstream tasks - no gps outputs. Will still use new image data for training.
09/10/2026	ModalAI Starling2	FA3WATHAK3	Sam's house, Loxahatchee, FL	Test GPS topic and beacon detector - gather beacon data	0.50	8:00 AM	Sunny, clear skies	Ran with solid config for all colors to get better ground truth beacon data - images appear to bright to detect color so need to modify settings.
09/11/2026	ModalAI Starling2	FA3WATHAK3	Sam's house, Loxahatchee, FL	Debug GPS not being subscribed correctly in custom code	0.50	9:00 AM	Clear skies	Determined qos setting issue in code. Modified script to correctly subscribe to GPS topic. Unable to continue testing with beacon due to weather.
09/28/2026	ModalAI Starling2	FA3WATHAK3	Sam's house, Loxahatchee, FL	Test flight with new controller and landing gear	0.50	10:50 AM	Clear skies	Completed flight pattern 1 with new landing gear and controller. Flight somewhat unstable and wifi - may need to recalibrate sensors. Ordered new wifi adapter for testing.

TABLE V.1: Flight test log for the Seabird drone platform.