Tigerfish: A Modular AUV Platform for RoboSub 2026

Mizzou Student Underwater Robotics Foundation
Technical Design Report

Lauren Bouckaert, Kaden Culbertson, Henry Bloch, Andrew Culbertson
Mizzou Student Underwater Robotics Foundation (SURF)
University of Missouri
Columbia, Missouri, USA
{lrbf7f, krc6m9, habcpm, anc5gg}@umsystem.edu

***Abstract*—For RoboSub 2026, Mizzou SURF is fielding *Tigerfish*, a complete ground-up redesign of our submarine platform. Rather than iterating on our previous vehicle (Jelly 2.0), the team prioritized a modular architecture that resolves three recurring failure modes from past campaigns: excessive positive buoyancy, difficult serviceability, and electrical noise between subsystems. Mechanically, Tigerfish is built on a one-inch aluminum extrusion frame with separate aluminum tubes for electronics and battery, bringing the vehicle near-neutral buoyancy without ballast and creating a modular mounting surface for future manipulators. Electrically, we adopted a commercial-off-the-shelf (COTS)-first strategy: a single COTS boost converter handles 19 V conversion for the Jetson Orin Nano, a COTS LiPo under-voltage cutoff provides battery safety, and an in-house power and sensor consolidation board solves the cable-management problems that plagued previous pool tests. The software stack is built on ROS 2, with YASMIN-based state management, OpenCV pipelines, and a YOLO-based deep learning detector for the gate, path, buoy, and bin tasks. A single-degree-of-freedom claw handles marker drops for the bin task, deliberately offloading positioning complexity to the vehicle's holonomic propulsion. A custom Unity simulator and Ansys Fluent CFD model provide a closed development loop for tuning PID controllers and validating behaviors before pool time. This report details the trade-offs between added complexity and reliability that drove these decisions, the resulting system design, and the testing strategy used to validate Tigerfish in advance of competition.**

I. Competition Strategy

A. Core Objective and Task Selection

Our objective for RoboSub 2026 is to autonomously complete every task in the competition except for the torpedo task. We have intentionally excluded torpedos this year in order to concentrate development effort on maximizing reliability across the remaining tasks.

A nominal run begins with Tigerfish being lowered into the water and autonomously submerging to operating depth. Using onboard cameras and the vision pipeline, the vehicle rotates on the yaw axis until the gate is visually detected, allowing us to attempt the coin flip. The robot estimates and corrects its position until it is completely aligned with the gate, then completes two rolls on the roll axis while steadily passing through the gate. Once through, the bottom cameras locate and follow the path marker leading to the slalom task. Vision is once again used as our primary sensor to stay centered to one side of the task and pass through each section.

After completing slalom, the vehicle locates the next path marker and navigates to the bin task. Using downward-facing cameras, the robot detects and aligns itself above the bins before deploying pre-loaded droppers through a modified claw mechanism. Once the bins task is complete, the vehicle transitions to locating the octagon task entirely through visual detection and navigation. After reaching and completing the octagon objective, the vehicle navigates back toward the gate area to conclude the run.

B. Complexity vs. Reliability Trade-off

The largest strategic decision this year was committing to a full vehicle redesign. We considered incrementally improving Jelly 2.0 and decided against it. Continuing with the existing platform would have preserved tested subsystems, but three structural issues were compounding year over year:

1) **Buoyancy.** Jelly 2.0's lightweight acrylic-and-PLA construction required an aluminum base plate plus add-on ballast, and the vehicle still fought to stay submerged. Tuning controllers on a vehicle that wants to surface masked real PID behavior.
2) **Serviceability.** The three-tier electronics stack required disassembling multiple layers to reach any single component, which compressed the diagnose-fix-test loop during testing days.
3) **Electromagnetic interference.** Motor lines and signal lines shared too much physical space, producing noise that previously degraded the hydrophone subsystem to the point that we have dropped acoustic pinger detection from this year's task set.

We accepted the short-term cost of a new hull, frame, and PCB layout in exchange for a platform that future Mizzou SURF teams can build on without fighting these same issues. This is the same long-runway argument that drives top teams: limited testing hours are better spent validating a clean architecture than patching a fragile one. To bound the risk, we held the software stack architecturally similar to the previous year's ROS-based system so the team is not redesigning both vehicle and code in the same cycle. The state-machine framework, vision pipeline, and PID structure carry over, with targeted upgrades described in Section II.

C. Capability-to-Task Mapping

Table I summarizes which Tigerfish subsystems are exercised by each chosen task. Vision is the common denominator and therefore receives the largest share of software effort.

TABLE I: Task–capability mapping for Tigerfish.

Task	Vision	Maneuvering	Actuation
Gate	Yes	Yes	No
Slalom	Yes	Yes	No
Bins	Yes	Yes	Yes
Octagon	Yes	Yes	Yes

II. Design Strategy

This section focuses on why each design choice was made and how it traces back to the strategy above. Detailed component specifications are provided in Appendix C.

A. Mechanical Design

The mechanical redesign was driven by the three failure modes listed in the previous section. Each subsystem-level choice maps to a specific failure we intended to eliminate.

1) Frame: Tigerfish (Fig. 1) is built around a rectangular frame of one-inch square aluminum extrusion. Compared to the acrylic-window structure of Jelly 2.0, the extrusion frame provides a stiffer load path between thrusters, adds the weight needed to approach neutral buoyancy without bolt-on ballast, and exposes a continuous T-slot mounting surface. The slot interface means future subsystems (a claw, a torpedo launcher, additional cameras) can be added without redrilling the hull. This directly addresses our second strategic goal: reducing the cost of future modifications.

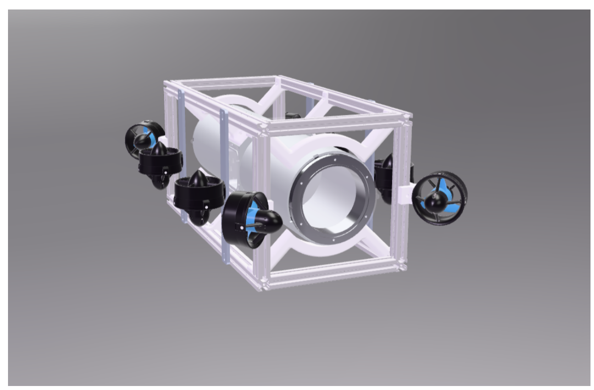

Fig. 1: Tigerfish full-vehicle render.

2) Pressure Hulls: We replaced the single acrylic main tube with two separate aluminum tubes: one for electronics and one for the battery, joined by a machined center mount that carries four acrylic viewing windows and the bulkhead penetrators for motors and the depth sensor. Splitting the hulls was an explicit choice for serviceability. Wiring runs are shorter, the battery can be swapped without touching the compute stack, and a fault in one tube does not require breaking the seal on the other. The aluminum also raises the weight enough that Tigerfish trims close to neutral without external ballast, eliminating the buoyancy struggle that compromised Jelly 2.0's controllability.

3) Tube-to-Frame Interface: The cylindrical tubes mount to the rectangular frame through custom ABS plastic saddles printed in-house. We iterated on this part across multiple revisions after early prints cracked under load at the mounting holes; the final geometry distributes load through a wider contact patch and survived our drop and torque checks (Section III-C). 3D printing was chosen for this part specifically because we expected to iterate on it. Using machined aluminum would have made each revision a multi-week turnaround; with 3D printing the loop is overnight. Two center saddles support the tube-to-tube junction and four end saddles anchor the tubes to the frame.

4) Battery Placement: The battery tube is mounted below the electronics tube. The LiPo pack is the single heaviest component in the vehicle, so placing it low brings the center of mass down to roughly the level of the thruster planes. This makes pitch and roll well-behaved without aggressive controller gains, and gives us margin to add top-mounted payloads in future years without re-trimming. The battery tube also houses a downward-facing camera used by the path-following state.

5) Manipulator: Single-DOF Claw: For the bin task we developed a single-degree-of-freedom claw that only opens and closes. We deliberately skipped a multi-axis arm. The instinct from land-based robotics is that more degrees of freedom means more capability, but underwater that calculus changes. A multi-axis arm needs waterproofed joint actuators, kinematic calibration, dynamic compensation for buoyancy torques as the arm moves, and a controller that does not fight the vehicle's station-keeping loop. None of that is impossible, but each item is a multi-week development effort, and we already had a vehicle to redesign.

Restricting the claw to a single DOF lets us push every other degree of freedom needed to position the marker onto the vehicle itself. Tigerfish is already holonomic in surge, sway, heave, and yaw, with controllable pitch and roll, so it can position the claw tip arbitrarily in the workspace relative to the bin using the propulsion system we are already building and tuning. The arm becomes a passive end-effector at the end of a six-DOF positioner; the vehicle does the control work. This collapses two control problems into one: tuning the vehicle's station-keeping controller is now also tuning the manipulation precision. Failures in arm hardware, such as a stuck servo or a wiring fault, degrade gracefully: the vehicle loses the ability to release markers but keeps the ability to move.

The claw mounts to the same T-slot frame extrusion the thrusters use, so it can be removed for tasks where it is not needed and replaced without modifying the hull. This same interface supports adding a more capable manipulator (a multi-DOF arm, a torpedo launcher) in future years without revisiting the structural design.

B. Propulsion

Tigerfish uses a vectored eight-thruster configuration: four horizontal and four vertical Blue Robotics T200s (Fig. 2). The horizontal thrusters sit at 45-degree corners to give holonomic surge, sway, and yaw control without dedicated sway thrusters; the vertical thrusters provide independent heave, roll, and pitch. The eight-thruster layout is intentionally heavier than the minimum needed for the 2026 tasks. The trade we accepted is the added current draw and ESC count in exchange for a flight envelope that supports future tasks (precision hovering above bins, attitude control during torpedo firing) without a propulsion redesign. Mounting on the T-slot frame lets us reposition any thruster without machining new brackets.

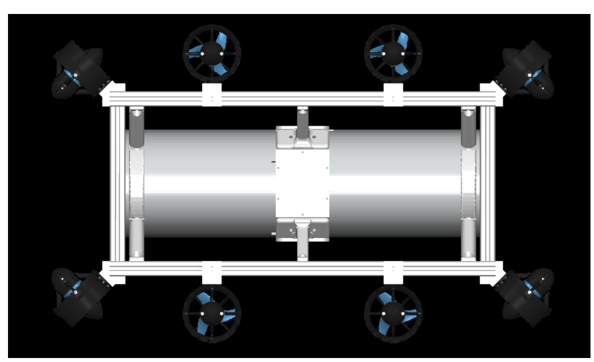

Fig. 2: Vectored eight-thruster layout.

C. Electrical Architecture

This year's electrical strategy is the largest change from prior designs. Past Mizzou SURF submarines pushed for fully custom PCBs across the entire electrical stack, including power distribution and DC-DC conversion. That approach repeatedly cost us time during the development cycle and produced boards that were not measurably better than parts available off the shelf. For RoboSub 2026 we inverted the default: commercial off-the-shelf (COTS) wherever it is functionally equivalent to a custom design, custom only where integration with the rest of the vehicle is the actual problem to be solved.

1) COTS Components: COTS parts handle the routine power and safety functions: a single boost converter for 19 V conversion, a COTS LiPo low-voltage cutoff for battery safety, and standard terminal blocks for bulk-power distribution to the ESCs and daughter rails. The LiPo cutoff replaces the previous comparator-and-AND-gate cutoff board; it achieves the same goal of hardware-independent under-voltage protection that cannot be defeated by a software fault or frozen Jetson, and we get the benefit of a tested, certified part with a known failure mode.

The COTS-first move freed roughly a semester of PCB design and validation effort. That time was reinvested into the mechanical redesign, controller tuning, and the vision pipeline, all of which are areas where the team's effort produces something not available off the shelf.

2) Custom Power and Sensor Consolidation Board: One piece of custom electronics on Tigerfish this year is an in-house power and sensor consolidation board (Fig. 3), designed by a team member to solve a problem that COTS parts could not.

The problem is not electrical, it is mechanical, and it was the single most common cause of lost pool-test time in previous years. With a loose collection of buck converters, sensors, and ad-hoc terminal blocks in the electronics bay, assembling the sub meant routing a tangle of free-floating wires past one another, and sensors regularly came unplugged during the final close-up. We would discover the disconnection only after the hull was sealed and the vehicle was in the water, costing an entire test cycle to recover.

The consolidation board addresses this directly. All sensor connectors and the routing for power distribution land on a single mechanically anchored PCB. The board retains the connectors itself; we do not rely on friction in floating Dupont headers, so a connector that was seated correctly stays seated through hull assembly. The board does not add electrical sophistication. It is intentionally simple, but it removes a recurring failure mode that no off-the-shelf part addresses. This is the criterion we now apply for any future custom electronics: the value has to come from integrating with the rest of the vehicle. Re-implementing a circuit that already exists as a module does not clear the bar.

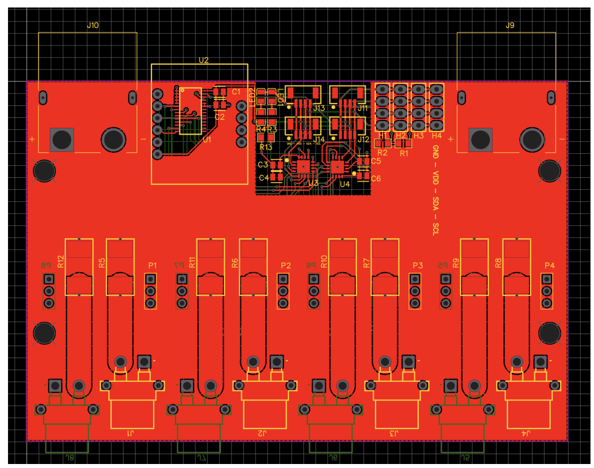

Fig. 3: Custom power and sensor consolidation board.

3) Forward-Looking Strategy: The COTS-first approach is a deliberate baseline. It is not a permanent ceiling. As the team's PCB design and validation experience grows, we expect to selectively reintroduce custom boards where doing so will measurably outperform COTS through tighter integration, lower part count, better fault behavior, or features no module provides. The consolidation board described above is the first of these. Future candidates include a dedicated motor-driver board if ESC integration becomes a bottleneck, and a higher-channel-count sensor interface if we add subsystems that exceed the COTS modules' I/O. Each candidate gets an explicit build-vs-buy decision; we do not default in either direction.

D. Software Stack

The software stack is built on ROS 2. ROS was chosen for the same reasons it underpins most competitive AUV programs: a node–topic abstraction maps cleanly to the sensor–processor–actuator pipeline of an AUV, and the existing library ecosystem (camera drivers, transforms, simulation interfaces) removes a large class of plumbing work. Figure 4 shows the planned architecture: sensor nodes (blue) feed processing nodes (green) that publish filtered estimates onto topics consumed by the state estimator (pink) and the task state machine (yellow) and output to nodes (grey). Additionally the main node is red as it has longer term memory.

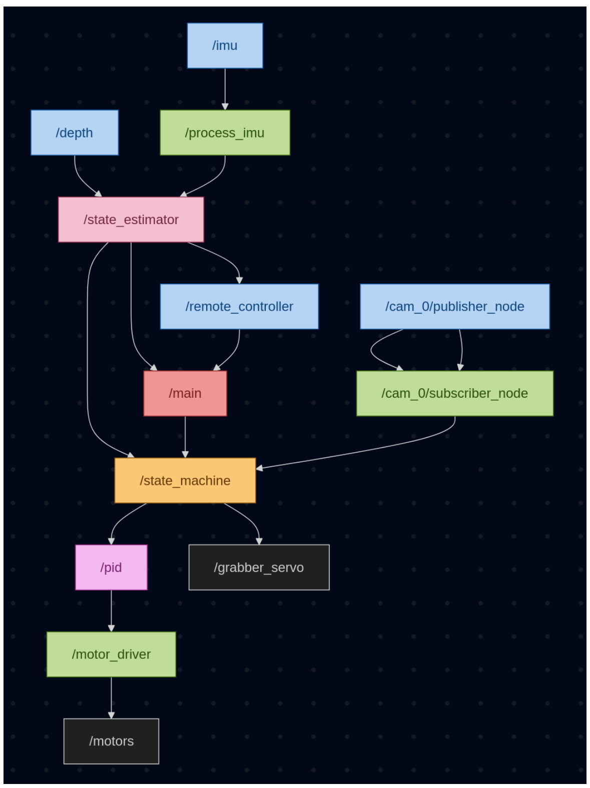

Fig. 4: ROS 2 software architecture for Tigerfish.

1) State Machine: We use YASMIN [6] as the state-machine framework. YASMIN's state and transition model is explicit and readable, which matters more to us than a graphical editor: most of our software contributors are newer members, and an explicit Python state definition lowers the barrier to contribution far more than a GUI would. YASMIN's blackboard, combined with ROS 2 nodes, gives a clean mechanism for sharing state between transitions, cameras, IMU, and (eventually) manipulator subsystems.

2) Perception: Detection of competition elements uses a hybrid pipeline. A YOLO-based deep learning detector [4], [5] handles the gate, buoys, and generally any object with rich visual structure. This avoids the brittleness of hand-tuned thresholds when lighting changes. The downward-camera path-follower uses classical OpenCV: an HSV mask isolates the orange floor markers, morphological operations [3] clean the mask, the largest contour is extracted, and a fitted line yields the heading correction fed back to the controller. We keep classical CV for the path task because the geometry is simple and the algorithm is easier to debug at the pool than a neural network is.

3) Control: Each degree of freedom (surge, sway, heave, roll, pitch, and yaw) is controlled by an independent PID controller. The output of each controller is translated into motor commands, and the combined outputs are summed to generate eight individual motor signals that drive the vehicle toward the desired state. An optional cascaded control structure is also supported, where the lower-level PID controller either receives direct velocity commands or receives velocity targets from a higher-level positional PID controller. PID gains are tuned first in simulation (Section III-A) and then refined in-water.

III. Testing Strategy

Testing time on a real submarine is the binding constraint for our team: pool access is shared, transport is non-trivial, and an in-water failure costs an entire test day. Our testing strategy is built around pushing as much validation as possible left of the pool, into simulation, onto benchtops, and into hardware-in-the-loop rigs, so that pool days are spent on what only the pool can answer.

A. Simulation-First Testing

Two complementary simulators bracket the problem. A custom Unity environment renders the vehicle, the pool, and the task props, and is the primary fixture for tuning PID gains, validating state-machine transitions, and producing synthetic camera footage for vision training. Custom shaders approximate underwater effects so that the synthetic frames are close enough to real camera output to be useful for training. The Unity scene also has a dedicated PID tutorial mode used to onboard new members.

In parallel, an Ansys Fluent CFD model gives us drag and added-mass estimates that feed back into the Unity physics model. This is the closed loop that makes Unity useful as a controller-tuning fixture: drag coefficients are not guessed, they come from CFD. We do not claim the simulator is a perfect analog, but the combination tightens the gap enough that the first in-water run of a new controller behaves recognizably like the simulated one.

B. Bench Testing

1) Software–Hardware Integration Bench: A motor testbench replicates the eight-thruster configuration with the vehicle's ESCs and a bench power supply. This lets us validate the PWM-output node, ESC arming sequence, and motor allocation math against real hardware while the actual sub is dry. Sensor input nodes (IMU filtering, smoothing) are validated on the bench against known motions before the IMU goes into the hull.

2) Electrical Bench Testing: Every PCB, custom or COTS, is brought up incrementally before it is integrated: power rails first with current limited at the supply, then digital lines, then the analog-to-digital chain. The COTS LiPo cutoff described in Section II-C2 is validated by sweeping a programmable bench supply across the cell-voltage range and verifying that the cutoff fires at the rated threshold; we treat the COTS part the same as a custom board for the purposes of safety validation, because the safety claim depends on the part working as advertised. The custom power and sensor consolidation board is bench-tested for connector retention as well as continuity. We deliberately yank on each connector under load to confirm that the board itself holds the connection, with no help from friction in the housings. We use an oscilloscope to verify the PCA clock signal and ADC timing where the digital path is sensitive to edges. The ADC subsystem (a MAX11057 four-channel simultaneous-sampling part) revealed a non-trivial bring-up issue during the spring semester: monitored pins on the Teensy cycled between three voltage levels instead of resolving the full 14 bits. We isolated the fault to the ADC driver code; the analog front end is fine. The fix is in progress and is documented as a tracked issue; it is not yet closed.

C. Mechanical Testing

Every newly fabricated mechanical part is fit-checked against the existing assembly immediately on receipt. Parts that interfere are modified and reprinted; this rapid loop is the main reason the tube-to-frame saddles were 3D-printed and not machined. We expected multiple revisions and got them. Sealed enclosures are vacuum/pressure leak-tested before any electronics go inside. After successful seal test, full electronics are installed and the assembled vehicle is trim- and stability-checked in a test tank to verify buoyancy and center of mass against the design target.

D. In-Water Testing

Pool days follow a fixed protocol: a pre-test bench check (thrusters, camera feed, IMU, depth sensor, telemetry, leak verification), a tethered manual-control verification of all eight thrusters in-water, then autonomous task runs in priority order. Conditions for every run (water temperature, ambient lighting, software version, controller gains) are logged so behaviors can be reproduced. A safety operator on the deck has sole responsibility for the kill switch and is not also managing software. We adopted this separation after early tests showed how easily that role gets dropped.

E. Outcomes Driving Strategy Adjustments

Two findings from this year's testing have already fed back into the design. First, the ADC issue described above pushed us to drop hydrophone-based acoustic detection from the 2026 task set. Shipping a half-validated subsystem was the alternative, and we treated it as unacceptable. This is a deliberate scope reduction in line with the complexity-vs-reliability trade in Section I. Second, early Unity-tuned PID gains were noticeably too aggressive when first ported to water because the simulator's drag model underestimated the real value at low velocities; this directly motivated the Ansys-driven update to the Unity physics described above.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, 2015.

[3] The MathWorks, Inc., "Types of Morphological Operations," MATLAB Documentation. [Online]. Available: https://www.mathworks.com/help/images/morphological-dilation-and-erosion.html

[4] K. Liu, L. Peng, and S. Tang, "Underwater object detection using TC-YOLO with attention mechanisms," *Sensors*, vol. 23, no. 5, p. 2567, Jan. 2023, doi: 10.3390/s23052567.

[5] 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.

[6] M. Á. González-Santamarta, "YASMIN: Yet Another State MachINe," GitHub repository, 2023. [Online]. Available: https://github.com/uleroboticsgroup/yasmin

[7] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot Operating System 2: Design, architecture, and uses in the wild," *Science Robotics*, vol. 7, no. 66, May 2022, doi: 10.1126/scirobotics.abm6074.

[8] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

Appendix A
Acknowledgements

Mizzou SURF gratefully acknowledges the support that makes this program possible. Our faculty advisor, Dr. Kristofferson Culmer, provides ongoing technical guidance and program advocacy within the College of Engineering. The Mizzou College of Engineering supplies laboratory space and access to the manufacturing equipment used to fabricate Tigerfish's custom aluminum and 3D-printed parts. SolidWorks and MathWorks provide software licenses for mechanical design and signal processing work, respectively. NVIDIA's developer program supports the onboard compute platform used for vision inference. We also thank the RoboNation organizers, judges, and volunteers who run RoboSub each year.

Appendix B
Outreach Activities

Mizzou SURF participates in outreach events throughout the academic year to recruit new members and promote underwater robotics and engineering education at the University of Missouri and in the surrounding community.

- **Get Involved Fair** (August 27): University-wide student organization recruitment event reaching several hundred incoming students.
- **New Student BBQ** (September 2): College of Engineering welcome event for incoming engineering students.
- **Organization Showcase** (September 10): Engineering student organization fair.
- **Caterpillar Build Room Tour** (September 12): Hands-on lab tour for approximately 30 prospective members.
- **E-Week Lab Exhibits**: Annual engineering week event showcasing student team projects.
- **Future Engineer Fridays**: Recurring program introducing high school students to engineering disciplines.
- **Jefferson Middle School Fair** (February 17, with the Academy of Engineering): STEM outreach event reaching 20–50 middle school students.

Appendix C
Safety

Tigerfish's safety architecture combines hardware-level and procedural safeguards.

Hardware Safety:

- **Battery cutoff.** A COTS LiPo low-voltage cutoff (Section II-C2) provides hardware-independent under-voltage protection on the LiPo pack, independent of any software fault on the Jetson.
- **Mechanical kill switch.** An external switch on the vehicle de-energizes the main power rail and is accessible without opening the hull.
- **Bulkhead penetrators.** All external connections use waterproof bulkhead connectors rated above operating depth; routine pressure/leak tests precede every wet test.

Procedural Safety:

- **Designated kill operator.** On every pool test, one team member is assigned sole responsibility for the kill switch and is not concurrently running software.
- **Pre-test checklist.** Every pool day begins with a fixed checklist: leak verification, thruster spin check, sensor feed verification, telemetry handshake, and battery state.
- **LiPo handling.** Batteries are stored, charged, and transported in LiPo-safe bags. Charge state is logged before and after every test.

Appendix D: Component Specifications

TABLE II: Tigerfish component specifications.

Component	Vendor	Model/Type	Specifications	Custom/Bought	Cost	Year
Hull plate	Midwest Steel	6061 Al	0.25 in × 16 in × 16 in plate, drilled for bulkheads	Custom	$108	2022
Battery tube	Custom	—	6 in polycarbonate tube, custom end caps	Custom	$160	2023
Thrusters (8×)	Blue Robotics	T200	8.2 / 6.4 lbf fwd/rev @ 12 V	Bought	$1,600	2025
ESCs (8×)	Blue Robotics	Basic ESC	30 A BLDC ESC	Bought	$320	2026
Power+sensor bd.	Custom	—	In-house power delivery & sensor consolidation board	Custom	—	2026
Boost Converter	Generic	Generic	14.8 V → 12 V DC-DC module	Bought	—	2026
LiPo cutoff	Generic	Generic	Low-voltage alarm + cutoff	Bought	—	2026
Battery	Turnigy	High-cap LiPo	20,000 mAh, 4S, 12C	Bought	$120	2026
Compute	NVIDIA	Jetson Orin Nano	Onboard inference platform	Bought	—	2026
Forward Cameras x2	Arducam	2MP IMX323	1080P, 30 fps, 2MP Low Light Cameras	Bought	$80	2026
Downward Cameras x2	Arducam	2MP IMX323	1080P, 30 fps, 2MP Low Light Cameras	Bought	$80	2026
Claw	Custom	—	Single-DOF (open/close) marker-drop claw, servo-driven, T-slot mounted	Custom	—	2026
Buoyancy Control	Generic	Generic	Pool Noodles	Custom	$15	2026
Frame	Generic	Generic	Aluminum extrudes, PLA	Custom	$405	2026
Waterproof Housing	McMaster, Blue Train Engineering, Midwest Steel	–	Aluminum tube, acrylic plates	Custom	$369	2025
Waterproof Connectors	Blue Trail Engineering	Various	–	Bought	$514	2025
Motor Control	Blue Robotics	Basic ESC	–	Bought	$240	2024
High Level Control	–	–	–	–	–	–
Actuators	ZOSKAY	–	40kg RC 6.8V 1/8 1/10 180 degree	Bought	$19	2026
Propellers	–	–	–	–	–	–
Converter	Generic	Generic	19V	Bought	$10	2025
Regulator	–	–	–	–	–	–
CPU	Nvidia	Jetson Orin Nano	–	Bought	$250	2026
Internal Comm Network	–	I2C	–	–	–	–
External Comm Interface	–	–	–	–	–	–
Compass	–	–	–	–	–	–
Inertial Measurement Unit (IMU)	Adafruit	BNO055	9-DOF Absolute Orientation	Bought	$35	2025
Doppler Velocity Log (DVL)	–	–	–	–	–	–
Manipulator	Custom	Custom	Custom	Custom	$50	–
Algorithms	–	Custom	–	Custom	–	–
Vision	Arducam	2MP IMX323	1080P, 30 fps, 2MP Low Light Cameras, x4	Bought	$160	2026
Acoustics	–	–	–	–	–	–
Localization –	Mapping	–	–	–	–	–
Autonomy	YASMIN	State Machine	Custom	Custom	–	–
Open Source Software	ROS2	Humble	–	Custom	–	–
Inter-Vehicle Communication	–	–	–	–	–	–

Continued on next page. . .

TABLE II: Tigerfish component specifications (continued).

Component	Vendor	Model/Type	Specifications	Custom/Bought	Cost	Year
Programming Language(s)	–	C++, Python	–	–	–	–