MARVL: Design and Competition Strategy for the 2026 Maritime RobotX Challenge

Team MARVL
Multi-Agent Robotics, Vision and Learning Lab
Singapore University of Technology and Design, Singapore
RoboCommand team identifier: SUTD

***Abstract*—Team MARVL from the Singapore University of Technology and Design enters RobotX 2026 with three vehicles run as one system from a single ground station: a BlueBoat surface vessel, a BlueROV2 underwater vehicle and a 22 inch quadrotor built in the lab. Every 2026 task needs at least two domains, so the strategy is to build the parts that several tasks share first, and once. Commercial hulls were chosen where the problem is solved, the aircraft was built to meet the 7 kg, floating, payload carrying requirement, and all vehicles sit behind one MAVLink adapter so the same mission code runs in model simulation, in ArduPilot software in the loop and on hardware. One orchestrator holds the vehicle, task and mission state machines, relays information between vehicles and is the only component on the judges' RoboCommand network. A capability model grades 19 capabilities by their evidence and multiplies them along each task tier. It sets the declared targets (Advanced for Safe Passage and Dynamic Incident Response, Core for the other two tasks) and ranks the remaining work, led by shared colour perception and two known software defects. Testing runs at three levels: 154 automated tests on every commit, a 25 stage mission against real firmware over an emulated radio before field sessions, and recorded pool and field sessions. The boat has run the readiness gate course autonomously, the aircraft has flown eight recorded flights including geofence and link loss tests, and the run start sequence has passed against RoboNation's own test server.**

I. Competition Strategy

A. Mission-Driven Strategy

MARVL fields USV, UUV and UAV platforms as a coordinated system for the four RobotX 2026 tasks. Because every task requires cooperation across multiple domains, development prioritises capabilities reused across missions: inter-vehicle information relay, colour and flashing-light perception, shared mission execution and UAV delivery. The design follows three principles. First, one shore-based orchestrator coordinates all vehicles through a common MAVLink interface. Second, commercial USV and UUV platforms are used where the platform problem is already solved, while the UAV is custom-built to satisfy the mass, flotation and payload constraints. Third, a competition tier is declared only when all required capabilities are supported by simulation, hardware or recorded-field evidence. Safety and readiness requirements are verified before mission capability is added.

B. Trade Study: Complexity against Reliability

Following the confidence based approach of the 2024 winner [1], the team made the trade between added capability and reliability explicit with a capability model. The handbook scoring section was unpublished when it was built, so the model estimates the chance of completing a tier and does not weight by points.

Each capability i is graded by the strongest evidence behind it, and each level maps to a confidence c_i: demonstrated on the vehicle (H, 0.9); verified against a real counterpart off the vehicle such as firmware in the loop, RoboNation's server or recorded imagery (V, 0.7); run in model simulation only (M, 0.5); partly built or carrying a known defect (B, 0.3); not started (N, 0.1). A tier needs every capability in its set S_t, so it is modelled as a series system [2]:

$$P_t = \prod_{i \in S_t} c_i. \tag{1}$$

The number of capabilities $n_t = |S_t|$ measures complexity. Each added capability multiplies in another factor below one: a larger system has more ways to fail. Table I grades the 19 capabilities today and after the planned tests, and Table II gives the tier estimates.

Declaration rule. For each task the team declares the highest tier in which every capability reaches level V or better after the planned tests and whose planned estimate is at least 0.3. At 0.3 per attempt, two attempts give $1 - 0.7^2 = 0.51$, an even chance of one success. The rule selects Advanced for Tasks 1 and 4 and Core for Tasks 2 and 3. The Advanced tiers of Tasks 2 and 3 need ten capabilities each, and even with every planned test passed their estimates stay near 0.15. Task 1 Advanced costs little over Core (0.321 against 0.397) because it swaps surface colour detection for top beacon detection plus a relay the team already has. No Disruptive tier is declared, because C18 and C19 would still exist only in model simulation.

Ordering the work. The first order gain from raising capability i to its planned level c_i' is

$$G_i = \sum_{t\,:\,i \in S_t} P_t \left(\frac{c_i'}{c_i} - 1 \right), \tag{2}$$

evaluated at today's P_t. The largest values are C1 (0.26), C17 (0.25), C18 (0.13), C4 (0.09), C3 (0.09), C6 (0.08) and C13 (0.08). Two of the top five are defects in existing software, each a few hours of work: an hour spent closing a defect in an existing capability returns more than an hour spent opening a new one. The tin grabber, flash decoding and circle detection each show zero gain alone, because their tiers are gated by other unstarted items, so the delivery chain is built and tested as one piece.

The model is coarse (independence, five levels, no points), but it makes the tier decision reproducible from evidence and ranks the remaining work.

C. Design Decisions and Their Trade-Offs

Table III records the architectural choices made for reliability or schedule, each expanded in Section II.

D. Risk

The main risks to the declared tiers, with their mitigations: perception, which holds five of the seven unstarted capabilities (one shared colour classifier, Sec. II-H); UUV water time, with its readiness package and DVL bottom lock still open (every behaviour flown against ArduSub in the loop first); aircraft mass, 0.15 kg under the limit (every addition weighed, Sec. II-G); and the course radio environment (testing on a fringe link profile, Sec. III-B).

II. Design Strategy

A. Systems Engineering Approach

The team follows the V model [4], [5], with verification at each level traced to a requirement. The requirements are held as data in the ground station. Handbook 3.1 is encoded as 63 readiness requirements with their section, domain and evidence type; 36 carry a measurable rule that the rehearsal simulator judges. Section 3.3 is encoded as an ordered step list per task and tier, which the mission executive walks. Section 3.4 is covered by tests that exchange real protocol buffer messages over a real MQTT broker, and sections 5.2 and 5.3 map to a safety table per vehicle. A handbook change therefore becomes a failing test, and each passing test traces to a clause.

B. System of Systems Architecture

Fig. 1 shows the architecture. The vehicles run ArduPilot [6] (ArduRover, ArduSub and Copter), each with a manual link to its own transmitter that works with the ground station off. The BlueBoat carries a WiFi HaLow (IEEE 802.11ah [7]) radio to shore, chosen for range at low data rate; the BlueROV2 is tethered to the boat, which carries its data; the aircraft uses 2.4 GHz telemetry.

A Python vehicle model, a firmware build in the loop and a real hull all implement the same adapter interface, and switching between them is one line of configuration. This is how the mission logic was developed and tested for three months before the vehicles were in the water.

TABLE I
CAPABILITIES, THE EVIDENCE BEHIND THEM ON 29 SEPTEMBER 2026, AND THE PLANNED TEST THAT WOULD RAISE EACH ONE

ID	Capability	Tiers	Evidence now	Now	Planned test	Plan
C1	RoboCommand reporting and run start	10	Six of six checks against RoboNation's rc-test server	V	Three vehicle session on the water	H
C2	USV guided navigation	8	Autonomous readiness gate run in the pool	H	None needed	H
C3	USV buoy circling	3	Software in the loop check fails in 2 of 4 runs	B	Revised circling (Sec. II-E) in the loop	V
C4	Colour light classification, surface camera	5	2024 detector, colour classes only	B	Recorded footage of 2026 style beacons	V
C5	UAV autonomous flight	4	Three recorded autonomous missions	H	None needed	H
C6	Top beacon detection, UAV camera	2	None	N	Recorded overflight of a top panel	V
C7	Inter-vehicle relay through the ground station	4	Route relay stage passes in the loop	V	Three vehicle session	H
C8	UUV DVL-aided position and velocity control	2	ArduSub in the loop; DVL configured, bottom lock untested in water	V	Pool session	H
C9	Pipeline acquisition and following	2	YOLOv11 on the Jetson, bench verified on image sets	B	Pool footage of a pipeline mockup	V
C10	Probe positioning on a light node	2	Probe in build	B	Pool, mock light node	H
C11	Flash sequence decoding	2	None	N	Recorded flashing sequences	V
C12	Docking approach	2	Course model only	M	Built docking bay	H
C13	Water jet on the window	2	None	N	Built docking bay	H
C14	Tin grab and release	2	Mount flown at 6.8 kg; grabber in build	B	Delivery flights	H
C15	Delivery circle detection, UAV camera	2	None	N	Recorded overflight of circles	V
C16	Break off, loiter, resume (Task 4 Core)	1	Suspend and resume pass in the loop	V	None planned	V
C17	Keep-out avoidance (Task 4 Advanced)	1	Handler raises an exception at its first step	B	Fix, then in the loop	V
C18	Moving object standoff (Task 4 Disruptive)	1	Resolves on a timer; no projected track	B	Projected standoff in the model	M
C19	UAV on station route updates	1	None	N	Model only	M

C. Inter-Vehicle Collaboration

Collaboration is a hub: the orchestrator writes what one vehicle found into the receiving vehicle's task parameters, advances that task's step and publishes the matching RoboCommand report. The flows are: UAV to USV with entry, exit and gate positions (Task 1 Advanced); UUV to UAV with the resource colour, and the destination at Disruptive (Task 2); USV to UAV with the delivery colour (Task 3); and RoboCommand to the affected domain for every Task 4 incident, answered with an acknowledgement on the wire.

The hub keeps one link per vehicle and one log. Its cost is latency: a relay crosses two radio hops, and on the fringe link profile (60 ms latency, 40 ms jitter) its worst case is about $2 \times (60 + 40) = 200$ ms, which is small against the 30 s and 60 s windows in which the Task 2 and Task 3 lights flash.

ArduPilot's default telemetry, more than thirty message types, caused heartbeat loss and emergency stops on the emulated fringe link. The adapter now requests two messages: `GLOBAL_POSITION_INT` at 4 Hz and `SYS_STATUS` at 1 Hz, plus the 1 Hz heartbeat. With MAVLink 2 framing (12 bytes of header

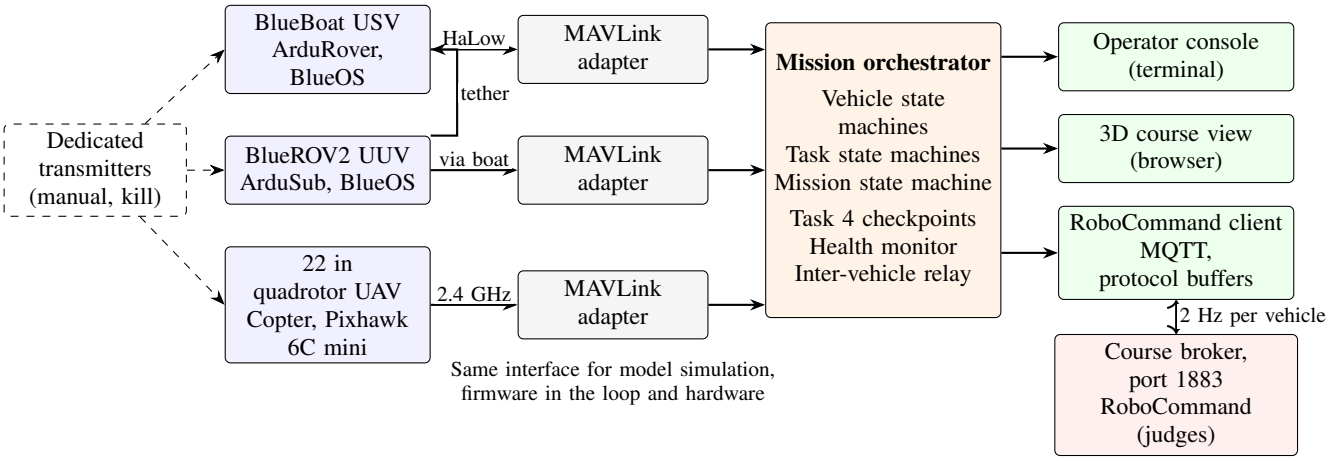


Fig. 1. System of systems architecture. All mission state lives in one orchestrator on shore. Only the RoboCommand client touches the judges' network (handbook 3.4). Dashed links are the manual control and kill paths, which work with the ground station switched off.

TABLE II
TIER ESTIMATES FROM (1), NOW AND AFTER THE PLANNED TESTS

Tier	n_t	Now	Planned	Decision
T1 Core	4	0.057	0.397	Fallback
T1 Advanced	6	0.012	0.321	**Declare**
T1 Disruptive	7	0.001	0.161	No, C19 at M
T2 Core	5	0.004	0.357	**Declare**
T2 Advanced	10	<0.001	0.128	No, $P < 0.3$
T3 Core	5	0.009	0.459	**Declare**
T3 Advanced	10	<0.001	0.164	No, $P < 0.3$
T4 Core	3	0.441	0.567	Fallback
T4 Advanced	3	0.189	0.567	**Declare**
T4 Disruptive	3	0.189	0.405	No, C18 at M

and checksum) that is

$$4(28{+}12) + (31{+}12) + (9{+}12) = 224 \text{ B/s} \quad (3)$$

per vehicle, or 3.1 kbit/s once 28 bytes of UDP and IP headers are added to each of the six packets. Three vehicles need 9.4 kbit/s, 3% of the 300 kbit/s fringe profile. A lost stream request is repeated within 3 s.

RoboCommand [8] uses MQTT 3.1.1 [9] on port 1883 with protocol buffer [10] payloads. The client validates the retained course, declares the run with a tier per task and the UAV geofence, heartbeats every vehicle at 2 Hz with per-vehicle sequence numbers, and checks that `RunStart` quotes the declaration's sequence number. Heartbeats carry the full field set on every report, with flight phase for the aircraft, because the same fields feed Singapore's network remote identification relay.

D. Mission Executive

The orchestrator keeps state in three machines, built on the `transitions` library [11] in the spirit of statecharts [12]: one per vehicle (OFFLINE, BOOTING, READY, AUTONOMOUS, TASK_EXECUTING, RETURNING, EMERGENCY_STOP), one per task attempt (INACTIVE, QUEUED, ASSIGNED, IN_PROGRESS, COMPLETE, FAILED, ABORTED) and one per run (IDLE, PRE_RUN_CHECK, RUNNING, EMERGENCY, COMPLETE). They are separate because a vehicle can be autonomous while its task is suspended and the mission is in an emergency; one machine would need a state for every combination. Each task holds the ordered step list for its tier, the smallest unit a mission can resume at.

On a Task 4 incident the orchestrator checkpoints the task (step, progress, score events earned), runs the incident as its own task and restores the checkpoint when RoboCommand clears it (Appendix A, Fig. 7). The checkpoint lives outside the task machine, which an interruption resets, so resuming is a data restore. A health monitor escalates sustained heartbeat loss (5 s)

TABLE III
ARCHITECTURAL TRADE-OFFS

Decision	More capable option	Simpler option	Chosen	Reason
Surface and underwater hulls	Custom hulls	BlueBoat and BlueROV2 through the granted platform programme	Commercial	Hull building would consume the season; the 2026 rules reward cross-domain autonomy
Aircraft	Commercial airframe	Custom 22 inch quadrotor	Custom	Payload mount, flotation and a 7 kg all-up limit had to be designed together
UUV positioning	Doppler velocity log aiding the autopilot filter	Depth hold only, tether-bounded to the boat	Doppler log, depth hold fallback	GUIDED control with bounded drift underwater; depth hold keeps the vehicle autonomous if bottom lock is lost
Mission software	Distributed nodes on each vehicle	One shore orchestrator over MAVLink	Orchestrator	One owner of mission state; runs unchanged against simulation, firmware in the loop and hardware
Inter-vehicle links	Direct vehicle to vehicle mesh	Hub through the ground station	Hub	Every relay is logged and mirrored to RoboCommand; one radio per vehicle
Telemetry	ArduPilot default stream set	Two messages per vehicle	Two messages	Default set saturated the emulated HaLow link (Sec. II-C)
Operator console	Graphical application	Terminal application	Terminal	Runs over a shell from the marquee; fixed layout under pressure
Buoy circling	Continuous line of sight guidance [3]	Waypoint hops	Hops, revised	Hops already worked; tolerance and spacing revised from analysis (Fig. 3)

to an emergency stop without waiting for the operator.

E. Surface Vessel: BlueBoat

The BlueBoat is a 1.2 m twin thruster catamaran running ArduRover on a Blue Robotics Navigator with BlueOS [13], driven in GUIDED mode with position targets. A Raspberry Pi runs BlueOS and a Jetson Orin NX runs perception on a ZED 2 stereo camera and a RoboSense RS-128 lidar; a WiFi HaLow radio links the boat to shore. The hull takes up to eight 4S Blue Robotics lithium ion packs, of which two are fitted for testing.

Kill switch chain, of original design. [14] Handbook 5.3.1 requires onboard and remote kill switches that disconnect propulsion power, including on loss of the remote link. The team designed an E-stop board for this (Fig. 2). The red mushroom switch on the hull and a PWM relay driven by the autopilot from the ExpressLRS [15] receiver are wired in series, so opening either one breaks the loop. The loop drives an optocoupler, which drives a BTS6163D smart high side switch, which energises the coil of an EV200 contactor in the battery feed to the ESCs. The kill therefore opens a mechanical contact in the power path, the switches carry only milliamps, and the smart switch has short circuit and over temperature protection and fails open. The relay is held closed only by a live PWM command, so a lost radio link, a stopped autopilot or a cut wire all de-

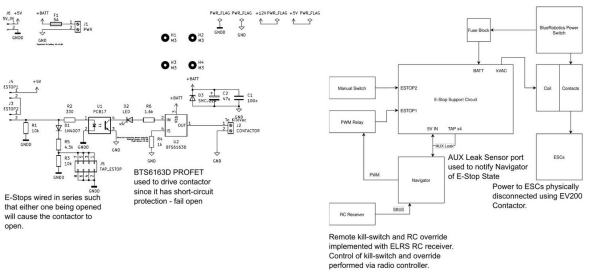


Fig. 2. USV E-stop board. Left: two E-stops in series drive an optocoupler and a BTS6163D high side switch, which holds the EV200 contactor closed. Right: the remote kill and manual override reach the board through the ELRS receiver and the Navigator.

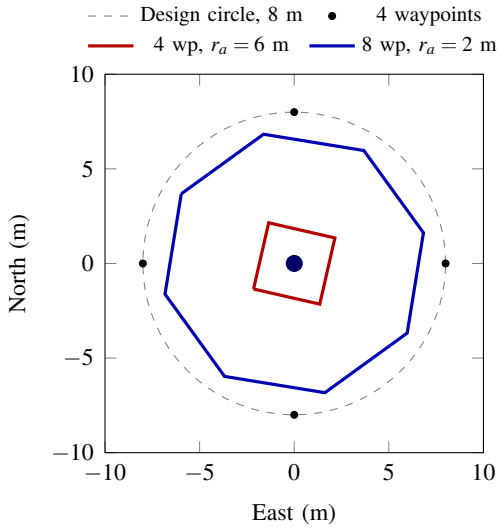


Fig. 3. Steady state path of waypoint circling around a buoy at the origin. Acceptance radius close to the circle radius collapses the loop to 2.5 m.

energise the contactor. Propulsion stops within 2 s of either switch opening. The onboard compute had been powered from the ESCs' 5 V outputs, so the power wiring was changed to put the contactor on the ESC feed alone: a kill stops the thrusters while compute and sensors keep running and logging. A tap from the loop goes to the autopilot's auxiliary leak input, so the autopilot knows it has been killed; it sets the status light to red and stays disarmed when power returns.

Labelled printed shrouds, four marked tow and lift points with a harness, a 360° mast light and a dedicated manual transmitter complete the checklist. The console treats manual takeover as an abort, since it ends scoring under rule 5.1.

Circling the entry and exit buoys. Task 1 requires a full clockwise circle of the entry buoy and a counterclockwise circle of the exit buoy. The first implementation sent four waypoints on an 8 m circle and advanced within 6 m of each, and failed in two of four runs in the loop. The geometry explains why: each leg starts r_a short of its waypoint, where r_a is the acceptance radius, so the loop contracts. For a point mass driving straight legs, four waypoints with $r_a = 6$ m settle into a loop of 2.5 m radius; $r_a = 4$ m gives 4.7 m and $r_a = 2$ m gives 6.5 m, while eight waypoints with $r_a = 2$ m give 7.0 m. With the original setting the path passes 1.8 m from the buoy, and a boat that overshoots a corner then misses the next arrival test, the timeout seen in validation. The revision uses eight waypoints and $r_a = 2$ m for 6.5 m of clearance (Fig. 3); line of sight guidance [3], [16], as in the NTU VRX stack [17], is the fallback.

For Task 3, docking and the water jet run as steps like any other task, so an interruption mid approach resumes at its step. Both are tested next against a built bay.

F. Underwater Vehicle: BlueROV2

The BlueROV2, in its Heavy configuration with eight T200 thrusters, runs ArduSub on a Navigator, and a Raspberry Pi running BlueOS carries video and MAVLink up the tether. Autonomy and pilot use separate endpoints.

Position underwater. ArduSub refuses GUIDED and POSHOLD whenever its filter has no position estimate, and underwater there is no GNSS. In software in the loop the estimate clears about a minute after boot; on the real vehicle it would never clear without a velocity or acoustic position source [18]. RoboCommand withholds the run start until every vehicle reports autonomous, so the protocol exposed this before

the water did. The vehicle therefore carries a Water Linked DVL A50 on the companion's Ethernet switch. Its bottom-track velocity feeds ArduSub's EKF3, which dead reckons a local frame and allows GUIDED body frame velocity and position setpoints. The adapter tries GUIDED first and falls back to depth hold (ALT_HOLD) if the filter has no position at arming, counting depth hold as autonomous for this vehicle only, so loss of bottom lock degrades control rather than stopping the run.

Position underwater is still dead reckoned: the log integrates velocity and drifts slowly, and in the depth hold fallback the vehicle steers on heading and time alone. Either way it must see the pipeline before it drifts off. With a heading error δ over a leg of length d, the lateral error is $d\sin\delta$. A camera at height h above the pipe with horizontal field of view θ sees a strip of half width $h\tan(\theta/2)$, so acquisition needs

$$d\sin\delta \le h\tan(\theta/2). \tag{4}$$

Taking $\delta = 5^\circ$, $h = 1$ m and $\theta = 80^\circ$ gives $d \le 9.6$ m. This sets the concept of operations: the boat, which knows its position, stations at the green buoy, and the UUV, held within its 6 m tether (handbook 5.3.2), descends beside the pipeline end. Its dead reckoned leg is a few metres, inside (4) even in the fallback, and it then follows the pipe visually.

The adapter offers two commands. `VELOCITY` sends body frame velocity with yaw rate, in GUIDED with the log or in depth hold as the fallback, and is the normal path. `RC_OVERRIDE` writes the channels a pilot's transmitter writes, mapped from a normalised command $u \in [-1,1]$ as

$$\text{PWM} = 1500 + 400u \ \mu\text{s}, \tag{5}$$

with offsets under the 50 μs thruster deadzone raised to 50 μs so small corrections are not lost. Unnamed channels are sent as *no change*, leaving them on the pilot's sticks. This path positions the magnetic probe, on a rigid forward mount aligned with the camera axis so it can be servoed visually into contact, on the red node.

Thruster power runs through a contactor interlocked to a hull switch and a wireless relay, which cuts the thrusters within 2 s while the companion computer and telemetry stay up. Manual override reaches the vehicle from an ExpressLRS receiver on the BlueBoat, down the tether. The tether is short enough that no winch is fitted, and the shrouds on both vehicles' thrusters guard against it tangling.

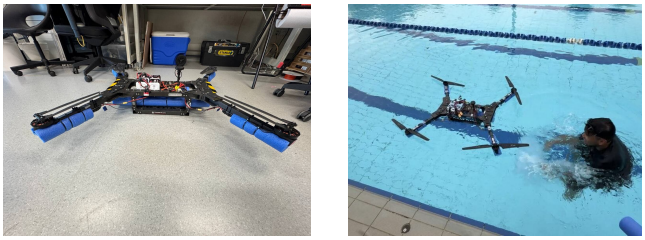


Fig. 4. The quadrotor with foam flotation along its arms (left), and the pool buoyancy test (right).

The vehicle is positively buoyant when killed and has an 80 m status light, marked tow and lift points, a vertical ULB-350 beacon mount and a fixed, buoyant 6 m tether. Since nothing may break the surface in a scoring run, depth hold is armed before the start.

G. Aerial Vehicle: Custom 22 Inch Quadrotor

The aircraft reads the buoy field, relays the route and carries tins. It is a 98 by 89 by 29 cm (129 cm diagonal) carbon quadrotor with 22.4 by 8 inch propellers on 320 KV motors, a Pixhawk 6C mini running Copter, and foam flotation (Fig. 4).

Mass budget. Handbook 5.3.3 limits the all up weight to 7 kg and requires the aircraft to carry an event-issued recovery light. The aircraft is 6.60 kg with batteries and flotation, the tin grabber adds 0.20 kg and a tin 0.05 kg, for 6.85 kg. That leaves 0.15 kg for the recovery light, so every change to the aircraft is weighed before it is fitted.

Flotation. To float at 6.85 kg the aircraft must displace at least 6.85 L. A 65 mm foam noodle displaces $\pi(0.0325)^2 \approx 3.3$ L per metre, so at least 2.1 m of noodle has to be under water before any margin, which is why it runs along all four arms and the body. The pool test confirmed it floats upright.

Power and endurance. Four 6S1P lithium ion packs of 4,500 mAh run in parallel, 18,000 mAh

TABLE IV
UAV BATTERY GATES AND FAILSAFES (VERIFIED STATUS IN APPENDIX A)

Trigger	Threshold	Response
Take-off gate	90% charge	No launch below this
Low battery	30%	Warning and return prompt
Critical battery	20%	Automatic land in place
Planned landing	30%	Missions planned to land with this reserve, about 15 min
Manual link loss	5 s	Land, disarm on touchdown
Telemetry loss	5 s	Land; pilot keeps control
Navigation loss	5 s	Land
Geofence	4 m buffer	Brake and hold at the fence
Ceiling	50 m AMSL	Hold at the declared ceiling
Mission abort	Console or pilot	Loiter or return to launch

at 22.2 V or about 400 Wh. Flight tests at 6.8 kg give about 18 minutes, which at 80% usable capacity is a mean draw of about 1.1 kW, or 160 W/kg. Missions are planned to land by 15 minutes, and four battery gates bound the risk (Table IV).

Limits and control. The declared ceiling is 50 m above mean sea level, 10 m under the handbook limit of 60 m. Descent is limited to the declared 2 m/s, and the 4 m geofence buffer makes the aircraft brake inside the site boundary rather than at it. The manual link, a RadioMaster ExpressLRS transmitter at about 0.08 W EIRP (limit 0.1 W), is independent of the ground station, and its emergency stop cuts the motors. The aircraft is registered with CAAS [19]; the pilot in command holds a CAAS unmanned aircraft pilot licence. Another additional pilot is nominated for licensing.

Payload. The 0.20 kg tin grabber is removable, on an underside mount powered from the main battery. The mount has been flown at 6.8 kg; the grabber is being finished.

H. Perception

The perception system uses YOLO-based models [20], [21] for real-time detection of mission objects. Training data are collected from custom-built replicas of RobotX targets and augmented across different viewpoints and lighting conditions. Object type and light colour are processed separately: YOLO detects the object, after which OpenCV classifies the colour within the detected region. This avoids treating every object–colour combination as a separate class and increases the training data available per object class.

USV detections are fused with lidar for range and relative-position estimation, while the UUV uses stereo vision for underwater depth estimation. Consecutive video frames are used to decode flashing lights and reject transient reflections. Object type, colour, confidence, range and relative position are passed to the navigation and mission-planning modules.

I. Software Tools

The ground station is 9.2k lines of Python 3.12: pymavlink in the adapters, `transitions` for the state machines, paho-mqtt and protobuf for the RoboCommand client, and Textual for the terminal console. ArduRover, ArduSub and Copter run on the vehicles and in the loop. Mosquitto and RoboNation's rc-test serve as the local broker and protocol reference. The course model and readiness rehearsal are 8.2k lines of JavaScript on Babylon.js. Perception uses YOLOv5 and the lab's ROS 2 camera capture nodes. Pilots and safety observers use Mission Planner and QGroundControl. Tests run under pytest and two headless Node harnesses.

III. TESTING STRATEGY

A. Approach and Time Allocation

Hardware time is the constraint: a pool booking is a morning, a boat session a day, and flights need a licensed pilot. The team tests at three levels, and nothing reaches a level before passing the one below:

1) *Every commit:* 154 automated tests, about 53 s, covering the state machines, suspend and

resume, the protocol end to end, the adapter, the link emulator and the course model.
2) *Before a field session:* the scripted mission against the real firmware over an emulated radio, 25 checks from a cold start to an emergency stop.
3) *In water and air:* test cards with a written objective and pass criterion, recorded start to finish so the recording doubles as readiness evidence.

Design and test time were split by building the test infrastructure early. The simulation layers and the link emulator were in place by 12 July, eight weeks into the season and before any vehicle ran autonomously, so every later feature was tested as it was written. The test code is 2.8k lines against 9.2k of source, a ratio of 0.30. The season timeline is in Appendix A (Fig. 8).

B. Simulation Testing

Four simulation layers each answer a different question. Python vehicle models check that the mission logic holds together, in seconds, inside the test suite. ArduPilot in the loop checks that the firmware accepts what the ground station sends. It runs the exact firmware the vehicles run, with ArduRover's motorboat model, ArduSub's vectored model and Copter's quad model, so arming checks, filter convergence, guided navigation and failsafes are real.

For the aircraft, the loop is used three ways before anything flies. The delivery profile to a 2 m platform (transit, descent over the 60 to 70 cm circle, release and climb out) is repeated until it completes without operator input. Manual link, telemetry and GPS loss are injected by cutting the simulated inputs, and simulated battery voltage is stepped down to fire each battery gate. Two of the three loss failsafes are not yet verified in flight, so a behaviour that misbehaves in the simulator never reaches the airframe.

A link emulator checks behaviour over a lossy, fading radio, and a 3D course model checks that the plan makes sense on the course and lets a judge see it.

The link emulator sits between the firmware and the ground station and adds latency, jitter, random loss, a bandwidth cap and periodic outages. Its profiles run from near (4 ms, 0.1% loss) through typical (15 ms, 1%, 1.5 Mbit/s) to fringe (60 ms with 40 ms jitter, 8% loss, 300 kbit/s, and a 2 s fade every 25 s). The fringe profile is the one that finds defects; it found the telemetry flood of Section II-C.

The validation scenario is one command that launches the fleet, runs a Task 1 mission with a Task 4 assistance request injected mid transit, and prints a pass or fail for 25 checks (Appendix A). Its exit code gates the build. The latest run passed 24 of 25; the failure is the circling defect of Section II-E.

The 3D course model (Appendix A, Fig. 9) is built in Babylon.js [22] from the handbook build specifications: ten buoys with side and top beacons, the survey buoys and pingers, a five section pipeline with five light nodes, three docking bays and two 2 m delivery platforms, set at the Promontory at Marina Bay. It can run the mission itself, mirror live RoboCommand traffic from a broker, read MAVLink from a firmware fleet or a real vehicle, or show the running ground station. A headless harness builds the scene without a screen, flies the mission and fails if a task does not complete, if RoboCommand messages arrive out of order, or if the underwater vehicle rises above its legal depth. A second mode rebuilds the four readiness courses of handbook 3.1 to the metre and judges 37 criteria as the rehearsal flies; all pass. VRX [23] was considered for camera simulation and is planned once perception is in the loop.

C. Component and Subsystem Testing

USV. The kill chain drawings and the propeller, tow point and control diagrams went into the readiness package. The boat then ran the handbook 3.1.2 gate course autonomously in the pool, from 3 m behind the first gate through both gates without contact (Fig. 5).

UAV. Twelve test cards, U1 to U12 in Appendix A, cover the failsafes, battery gates, fence, ceiling, link margin, payload, flotation and wind envelope. Each is run on the bench or in the simulator before it is flown, and the ones that risk

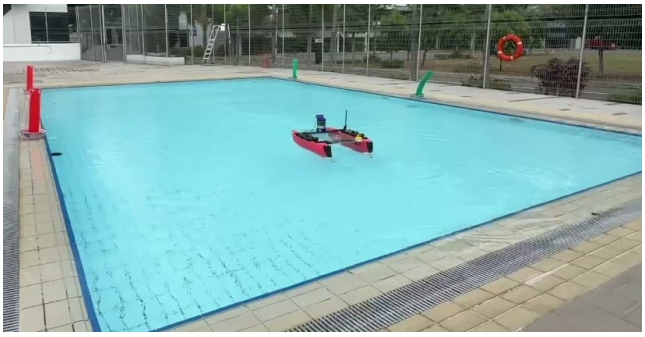


Fig. 5. The BlueBoat running the readiness gate course autonomously in the pool, between red and green gate buoys.

the airframe are flown last, low and over a clear area. The manual link failsafe, the geofence hold, the payload mount at 6.8 kg and pool flotation are verified; the telemetry and navigation failsafes and the ceiling hold are configured and waiting for flight verification.

UUV. Depth hold autonomy, the arming fallback and both command paths pass against ArduSub in the loop, and the lab's ROS 2 node drives the real vehicle on the same override channels. On the bench, velocity commands drive the correct thrusters and the kill contactor isolates the thrusters within 2 s with the companion still live. In the pool, the vehicle was driven manually through the tether, confirming watertightness, neutral trim and positive buoyancy on disarm (Table VIII).

Communications. The run start sequence of handbook 3.4.10 was run against RoboNation's own rc-test server [8], with Mosquitto [24] as the broker and three vehicles declared. All six checks passed and the server's log, written by RoboNation's decoder, was submitted for readiness.

D. Field Testing Performed and Its Impact

Field work to date is concentrated in the readiness campaign of August 2026: the BlueBoat autonomous gate run in the pool; three manual and three autonomous flights of the published pattern at an open field in Tuas, with a link loss flight and a geofence flight; a pool buoyancy test; and a bench motor test. The recorded flights total 15.0 minutes (7.7 manual, 6.0 autonomous, 1.3 safety) across eight flights. A crash during early PID tuning damaged the airframe; it was repaired and retuned, and tuning is now flown low over a clear area with the pilot ready to take over.

TABLE V
VERIFICATION AGAINST THE DECLARED TIERS

Tier	Verification	Status
T1 Adv.	Pool run (C2); revised circling in the loop (C3); top beacon footage (C6); UAV to USV pair (C7)	C2 done; rest planned
T2 Core	Pool survey on a mockup with a mock light node (C8 to C10); indicator footage (C4)	Planned
T3 Core	Built bay: dock and jet (C12, C13); indicator footage (C4)	Planned
T4 Adv.	Keep-out fix, then validation in the loop (C17)	Fix pending
All	rc-test sequence (C1); three vehicle session	C1 at V

Testing changed the design in six places:

- The UUV has no position underwater without its DVL, so depth hold was kept as an autonomous fallback and the survey leg is kept inside the dead reckoning bound of (4).
- The default telemetry flooded the fringe link, so the adapter requests two messages and re-requests them when they stop.
- Arming takes 20 to 40 s of filter settling, so arming retries for 90 s and mode is confirmed for 60 s after arming.
- The circling check fails intermittently, which led to the circling analysis of Section II-E and the revised circling.
- The 2024 detector transfers only in its colour classes, which led to the shared colour stage.
- The first simulated UUV mission surfaced the vehicle to pass a message, which breaches handbook 5.3.2, so the request now goes up the tether.

E. Integrated System Testing

Integration is tested in pairs before the three vehicle run. In the UAV to USV pair, the route the aircraft reads must be accepted by the boat's adapter and change its transit plan (Task 1). In the UUV to UAV pair, the resource colour must reach the aircraft and RoboCommand with no manual

forwarding (Task 2). The USV to UAV delivery is run end to end with the boat docked, to exercise the timing between docking and delivery (Task 3). The boat and UUV are also run together to check tether clearance from the shrouds and that video and MAVLink fit through the tether and HaLow link at once. Every pair runs at the competition telemetry rates, so latency and loss are measured under realistic load, and the three vehicle run is attempted only after all three pairs pass. One three vehicle session reporting to a local broker closes the plan. Table V maps each declared tier to its verification.

ACKNOWLEDGEMENTS

We thank Prof. Malika Meghjani, faculty advisor, for directing this entry and acting as airworthiness contact, and SUTD for the workshop, pool and field time. Blue Robotics provided the BlueBoat and BlueROV2 through the RobotX granted platform programme. RoboNation published the RoboCommand schema and rc-test server used for the communications submission. Garuda Robotics reviews the aircraft and pilot packages and runs pilot licensing. The lab's 2024 RobotX group built the buoy detector and lidar fusion reused here. Team Bumblebee (National University of Singapore) shared their communications submission as a format reference.

REFERENCES

[1] Team Minion, Embry-Riddle Aeronautical University, "Minion: Design and competition strategy for the 2024 Maritime RobotX Challenge," Technical Design Report, RoboNation, 2024.

[2] P. D. T. O'Connor and A. Kleyner, *Practical Reliability Engineering*, 5th ed. Chichester, UK: Wiley, 2012.

[3] T. I. Fossen, K. Y. Pettersen, and R. Galeazzi, "Line-of-sight path following for Dubins paths with adaptive sideslip compensation of drift forces," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 820–827, 2015.

[4] K. Forsberg and H. Mooz, "The relationship of system engineering to the project cycle," in *Proc. 1st Annual Int. Symp. National Council on Systems Engineering (NCOSE)*, 1991, pp. 57–65.

[5] D. D. Walden, T. M. Shortell, G. J. Roedler, B. A. Delicado, O. Mornas, Y. Yip, and D. Endler, Eds., *INCOSE Systems Engineering Handbook*, 5th ed. Hoboken, NJ, USA: Wiley, 2023.

[6] ArduPilot Development Team, "ArduPilot: Rover, Sub and Copter autopilot firmware," [Online]. Available: https://ardupilot.org, 2026.

[7] IEEE, "IEEE standard for information technology, part 11: Wireless LAN MAC and PHY specifications, amendment 2: Sub 1 GHz license exempt operation," IEEE Std 802.11ah-2016, 2017.

[8] RoboNation, "RoboCommand: RobotX 2026 communications schema and test server," GitHub repository. [Online]. Available: https://github.com/robonation/robocommand, 2026.

[9] A. Banks and R. Gupta, "MQTT version 3.1.1," OASIS, OASIS Standard, 2014.

[10] Google, "Protocol buffers," [Online]. Available: https://protobuf.dev, 2026.

[11] pytransitions contributors, "transitions: A lightweight, object-oriented state machine implementation in Python," GitHub repository. [Online]. Available: https://github.com/pytransitions/transitions.

[12] D. Harel, "Statecharts: A visual formalism for complex systems," *Science of Computer Programming*, vol. 8, no. 3, pp. 231–274, 1987.

[13] Blue Robotics, "BlueBoat, BlueROV2 and BlueOS technical documentation," [Online]. Available: https://bluerobotics.com, 2026.

[14] SUTD Organisation of Autonomous Robotics, "Emergency-stop support circuit," [Online]. Available: https://github.com/SOAR-UWU/estopsupport.

[15] ExpressLRS contributors, "ExpressLRS: Open source radio control link," [Online]. Available: https://www.expresslrs.org.

[16] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*, 2nd ed. Chichester, UK: Wiley, 2021.

[17] Tinker Twins, "SINGABOAT-VRX: Solution to the Virtual RobotX competition," GitHub repository. [Online]. Available: https://github.com/Tinker-Twins/SINGABOAT-VRX, 2022.

[18] L. Paull, S. Saeedi, M. Seto, and H. Li, "AUV navigation and localization: A review," *IEEE Journal of Oceanic Engineering*, vol. 39, no. 1, pp. 131–149, 2014.

[19] Civil Aviation Authority of Singapore, "Air navigation (101, unmanned aircraft operations) regulations 2019," Singapore, 2019.

[20] 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.

[21] G. Jocher *et al.*, "ultralytics/yolov5," Zenodo, doi: 10.5281/zenodo.3908559, 2020.

[22] Babylon.js contributors, "Babylon.js: A real time 3D engine," [Online]. Available: https://www.babylonjs.com.

[23] B. Bingham, C. Agüero, M. McCarrin, J. Klamo, J. Malia, K. Allen, T. Lum, M. Rawson, and R. Waqar, "Toward maritime robotic simulation in Gazebo," in *Proc. OCEANS 2019 MTS/IEEE Seattle*, 2019, pp. 1–10.

[24] R. A. Light, "Mosquitto: Server and client implementation of the MQTT protocol," *Journal of Open Source Software*, vol. 2, no. 13, p. 265, 2017.

APPENDIX A
TEST PLAN AND RESULTS

Scope

The objective is to show, before the event, that every capability in a declared tier (Table II) works at level V or better, and that every safety behaviour required by handbook 5.2 and 5.3 works on the vehicle. The test cases are the 25 validation checks (Table VI), the 37 rehearsal criteria, the UAV cards U1 to U12 (Table VII), the UUV cards (Table VIII), and the planned water and flight sessions of Table I. Not included: perception accuracy on the real course, which cannot be measured before the event; long endurance runs to battery exhaustion; and RF compliance measurements beyond the declared settings. Delivery circle detection accuracy is evaluated separately by the computer vision sub-team on its dataset.

Schedule

Software testing runs continuously from May.Readiness testing for the USV, UAV and communications closed with their submissions on 31 August 2026; the UUV package is completed alongside its pool sessions. From October the order is: the two software defects (C3, C17) and a clean 25 of 25 validation run; UUV pool sessions for DVL bottom lock, station keeping and a pipeline mockup; perception footage of beacons, indicators and circles; delivery flights; the docking bay session; the pairwise integration runs; and one three vehicle session before shipping. UAV testing is scheduled to close early enough to allow one full repair cycle before shipping.

Resources and Tools

The pool at SUTD for the boat and the ROV; an open field in Tuas for flights; the licensed pilot and a safety observer for every flight; the ground station laptop with the firmware fleet, the link emulator, Mosquitto and rc-test; Mission Planner and QGroundControl; ArduPilot dataflash logs downloaded after every flight; and an external camera recording every session. One battery set is cycled per test day, and packs are not discharged below the critical gate outside a gate test. Flight logs are downloaded and archived before the aircraft flies again. Spare propellers and arm hardware are held to shorten repairs. For the UUV, a pilot who also handles the tether and a telemetry operator attend every pool session.

Environment

Simulation runs on the ground station laptop and is not weather dependent. Pool tests are calm water with no current. Flights are within visual line of sight under the declared limits (50 m AMSL ceiling, 500 m range, 2 m/s descent, 8 m/s sustained and 10 m/s gust wind), with the geofence loaded with its 4 m buffer before every flight. Competition conditions that cannot be reproduced before the event are the salt water, current and radio environment of Marina Bay. Only the wind envelope test (U12) is flown near the wind limit, on days forecast inside it; all other flights are in light wind. The ditching test needs calm water.

Risk Management

Airframe damage is the largest test risk: tests that can fail on the bench are done props off first, and risky flights are flown last, low and over clear ground with spares on hand. Weather and access risk is reduced by keeping simulation and bench work independent of the field so field days are used only for what must fly or float. The UUV is only put in the water after the same behaviour has passed in the loop. Batteries are not launched under 90%, and packs that show degradation are retired. The repaired airframe is inspected before every flight. Field and pool bookings carry slack for one reschedule per phase, and the test order is kept flexible within each booking. The ditching test is flown without payload, with flotation verified in the pool beforehand and a recovery line on hand.

Results

The manual link failsafe (U1) and the payload mount at 6.8 kg (U10) were flown on 29 August 2026 and have been reconfirmed on later flights. The readiness rehearsals judged 37 criteria across the four readiness courses (USV 8, UUV 11 including the tether variant, UAV 13, communications 5), all passing. The rehearsal of the UAV

TABLE VI
VALIDATION SCENARIO AGAINST THE REAL FIRMWARE

#	Check	Result
1 to 3	Fleet ready with position; pre-run gate; mission running	Pass
4 to 6	RoboCommand connected; course received; run declaration	Pass
7 to 9	All armed and autonomous; run start; heartbeats reaching RoboCommand	Pass
10 to 12	UAV airborne; task dispatched; vehicles executing	Pass
13 to 18	Task 4: suspend, checkpoint at transit step, acknowledgement, readiness report, resolution, resume from checkpoint	Pass
19	USV guided waypoints including buoy circling	Fail
20 to 23	Safe passage report; task complete; mission complete; score events preserved	Pass
24, 25	USV return to launch; emergency stop disarms all	Pass

limit tests passes only when the clamp is seen to act: the aircraft is commanded 15 m past the fence and 20 m above the ceiling and must hold both for 2 s (Fig. 6). The communications run declared three vehicles; the fleet was held in manual for 60 heartbeats and then reported autonomous for 63, and RoboNation's server released run 1 on the third autonomous report.



Fig. 6. Readiness rehearsal of the UAV ceiling test: commanded 20 m above the declared ceiling and held at it.

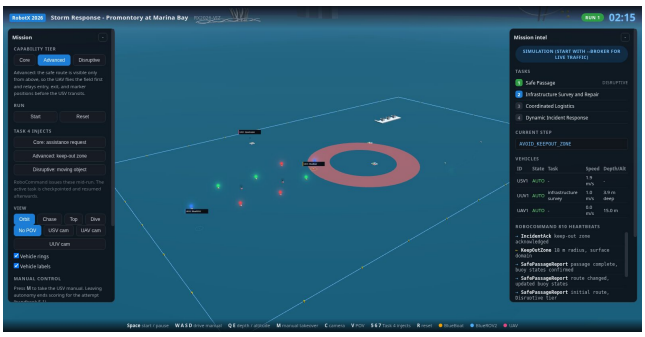


Fig. 7. Task 4 Advanced in the course model: RoboCommand issues a keep-out zone during the safe passage transit, the console acknowledges it, and the task resumes at the step it stopped on.

TABLE VII
UAV TEST CARDS AND RESULTS

#	Test	Method	Pass criterion	Status
U1	Manual link failsafe	Switch the transmitter off in flight	Land within 5 s of last input, disarm on touchdown	Verified in flight
U2	Telemetry loss failsafe	Cut the telemetry link, bench then low flight	Land; pilot keeps manual control	Configured; flight pending
U3	Navigation loss failsafe	Position loss in the simulator, then flight	Land after 5 s without position	Configured; flight pending
U4	Emergency stop	Props off on the bench, then low hover	Motors stop on command	Verified on the ground
U5	Battery gates	Bench discharge, then flight	Gates fire at 90, 30 and 20%	Thresholds set; end to end pending
U6	Geofence	Command past the polygon	Brake and hold at the fence	Verified in flight; recovery light mount fitted, light pending
U7	Ceiling	Command above the fence altitude	Hold at 50 m AMSL	Configured; flight pending
U8	Link margin	Flights at increasing range	Solid control to 500 m	Pending
U9	Tin grabber	Bench cycles, then hover pick and place	Reliable grab and release, no power sag	In build
U10	Payload mount	Fly at 6.8 kg	Stable hover and landing	Verified in flight
U11	Flotation	Pool test, then controlled ditching	Floats upright, recoverable	Pool verified; ditching pending
U12	Wind envelope	Flights at increasing gusts	Stable in 8 m/s sustained, 10 m/s gusts	Not yet tested

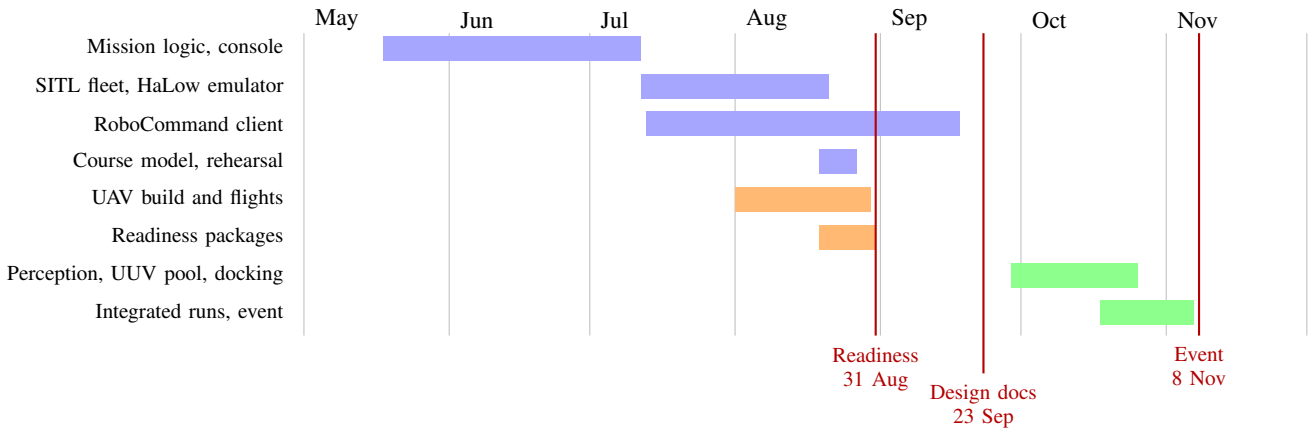


Fig. 8. Season timeline. Blue: software and simulation, dated from the repository history. Orange: hardware and readiness. Green: planned.

TABLE VIII
UUV TEST CARDS AND RESULTS

#	Test	Pass criterion	Status
B1	Bench command	Velocity commands drive the correct thrusters	Verified, Aug.
B2	Bench perception	YOLOv11 at 25 fps or more, under 50 ms, on the Jetson	Verified, Aug.
B3	Manual pool drive	Full response, neutral trim, positive buoyancy on disarm	Verified, Sep.
B4	Kill contactor	Thrusters isolated within 2 s, companion live	Bench verified; pool pending
B5	DVL bottom lock	Stable bottom-track velocity; EKF3 healthy	Pending
B6	Station keeping	Drift under 0.25 m over 60 s in GUIDED	Pending
B7	Pipe tracking	Lock held along a 5 m mockup	Pending
B8	Fault node	Stops and centres on the red node	Pending
B9	Tethered pair	Tether clear of shrouds; no heartbeat loss	Pending

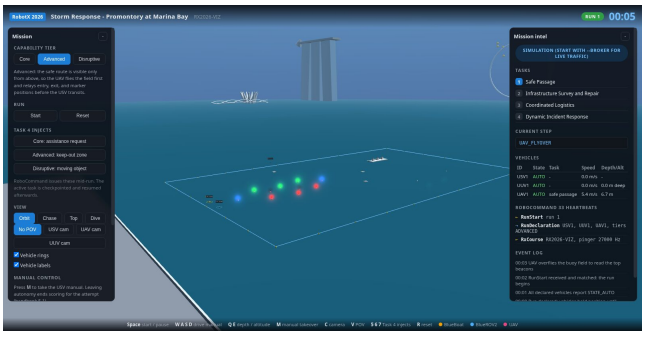


Fig. 9. The course model at Marina Bay with the Task 1 buoy field, driven by the same step lists as the orchestrator. The right panel shows the RoboCommand traffic a real link would carry.

APPENDIX B
COMPONENT LIST

TABLE IX
COMPONENTS BY VEHICLE

System	Component	Model or type	Source	Reason
USV	Hull and thrusters	Blue Robotics BlueBoat	Granted platform	Proven hull with ArduRover support
	Flight controller	Navigator with BlueOS, ArduRover	Blue Robotics	Standard BlueBoat electronics
	Companion and perception	Raspberry Pi with BlueOS; Jetson Orin NX	Blue Robotics; purchased	Perception on board
	Sensors	ZED 2 stereo camera, RoboSense RS-128 lidar	Purchased	Buoy detection and ranging
	Batteries	Up to eight 4S lithium ion, two in testing	Blue Robotics	Endurance scales with packs
	E-stop board	PC817, BTS6163D, EV200 contactor	Custom	Mechanical power break, fail open
	Manual and remote kill link	ExpressLRS receiver, PWM relay	Purchased	Independent of the ground station
	Shore link	WiFi HaLow radio	Purchased	Range at low data rate
	Shrouds, tow points, status light	Printed shrouds, marked harness, mast light	Custom	Handbook 5.2
UUV	Vehicle	Blue Robotics BlueROV2 Heavy, eight T200, ArduSub	Granted platform	Proven vehicle with ArduSub support
	Companion	Raspberry Pi with BlueOS	Blue Robotics	Video and MAVLink up the tether
	Camera	RTSP, 1280 by 720 at 15 fps	Blue Robotics	Pilot video
	Navigation	Water Linked DVL A50	Purchased	Bottom-track velocity to EKF3
	Compute and camera	Jetson Orin NX; Intel RealSense depth camera	Purchased	YOLOv11 pipeline and light detection
	Power	Two 4S lithium ion packs	Blue Robotics	Compute apart from thruster noise
	Beacon mount, magnetic probe	Vertical ULB-350 mount; probe	Custom	Handbook 5.3.2; Task 2 repair
	Kill and manual link	Contactor with hull switch and wireless relay; ELRS via the boat	Custom	Handbook 5.3.2
UAV	Airframe	22 inch carbon quadrotor	Custom	Payload, flotation, 7 kg limit
	Flight controller	Pixhawk 6C mini, Copter	Purchased	ArduPilot failsafes and fence
	Propulsion	320 KV motors, 22.4 by 8 in two blade propellers	Purchased	Lift at 6.8 kg
	Batteries	Four 6S1P lithium ion, 4,500 mAh	Purchased	About 18 min at 6.8 kg
	Manual link	RadioMaster Pocket, ELRS, RP3 V2 receiver	Purchased	Dedicated controller, 0.08 W EIRP
	Flotation, grabber	Foam noodles; tin grabber, 0.20 kg	Custom	Float after ditching; delivery
Shore	Ground station	Python orchestrator, console, RoboCommand client	Custom	One owner of mission state
	Simulation	ArduPilot in the loop, link emulator, 3D course	Custom and open source	Test before water time
	Perception	YOLOv5s buoy detector	2024 lab work	0.945 mAP@0.5