STONY BROOK ROBOTICS TEAM

FLOAT SPECIFICATION

MECHANICAL

The body of our float is built using a clear PVC tube, which we chose for its visibility, availability, and low cost. The transparency of the tube allows for easy inspection of internal components and detection of leaks during operation. The tube interfaces with custom-machined O-ring endcaps via a 4-axis CNC service (JLC CNC).

The buoyancy engine operates by using a peristaltic pump to intake and remove water from a soft bladder. We chose a peristaltic pump as they are bidirectional so it can push water in both directions. When we intake water, it increases the float's density and makes it sink. We can make it neutrally buoyant or float again by releasing water.

Figure 1: CAD Rendering of the Float

Figure 2: 1A Mini Blade Fuse

Figure 3: 12V 2000mAh NiMH battery

FLA in water [motor off]	0.1A
FLA in water [motor on]	0.7A
Fuse size selected	1A

Figure 4: Float Fuse Selection

ELECTRICAL

The float is powered by a 12V 2000mAh NiMH battery, providing enough current to run all onboard systems with a maximum draw of just 0.7A, nearly three times less than the battery's rated output. In testing, the float consistently completed five full missions per charge, exceeding the two required for competition. A smart charger is used for safe recharging, and the battery is stored in a fire-resistant bag when not in use. An inline 1A fuse protects the PCB from overcurrent.

All electronics interface through a custom PCB designed in Altium Designer. The board steps down the 12V input to 5V using a linear regulator and connects to the depth sensor, peristaltic pump, and motor driver. The motor is controlled by an H-Bridge with PWM input, fault protection, and current regulation. At the core of the system is an Adafruit ESP32 Feather V2, which manages control logic and telemetry. It features a USB-C port for programming, an RGB status LED, and a w.FL connector for an external WiFi antenna, enabling wireless communication with a surface laptop via 2.4GHz WiFi. Depth measurement is handled by the Blue Robotics Bar02 sensor, offering reliable readings up to 10 meters with 0.16 mm resolution, enabling precise vertical control during profiling missions.

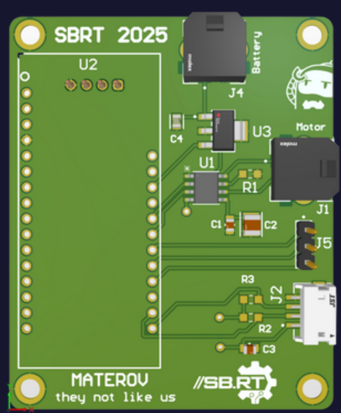

Figure 5: Render of the Float PCB

SOFTWARE

Communication

The ESP32 runs a custom UDP-based reliability layer: each packet carries a sequence number and must be acknowledged by the surface; unacknowledged packets are retransmitted. When the float breaks the surface and WiFi signal strength rises above –70 dBm, any stored measurements are POSTed in batches over HTTP to the laptop's /depth REST endpoint.

Control Logic

A PID loop on the Bar02 depth sensor drives the peristaltic pump via proportional (Kp), integral (Ki), and derivative (Kd) gains. The controller modulates bladder volume (max 250 mL) to hold a depth of 2.5m ± 0.125m—positive output floods the bladder to descend, negative vents to ascend, and outputs within ±10 stop pumping to avoid oscillation.

Safety & Velocity

To avoid over-pressurization, the float is always sealed with the bladder fully inflated. If pressure builds internally, the end caps are designed to pop off as a fail-safe, per competition safety guidelines. Descent and ascent speeds are capped to prevent overfilling the bladder or stalling the pump when the bladder is full.

Post-Mission Visualization

After retrieval, a Python/Plotly script loads the collected data, sorts by time, and then generates the depth vs time graph. The graph includes horizontal lines at y=2, y=2.5, and y=3 to visually show the expected depth, as well as the margin of error that the float is allowed to have, which is ± 0.5m as per the competition rules. Hovering over a data point displays the time and depth corresponding to that point.

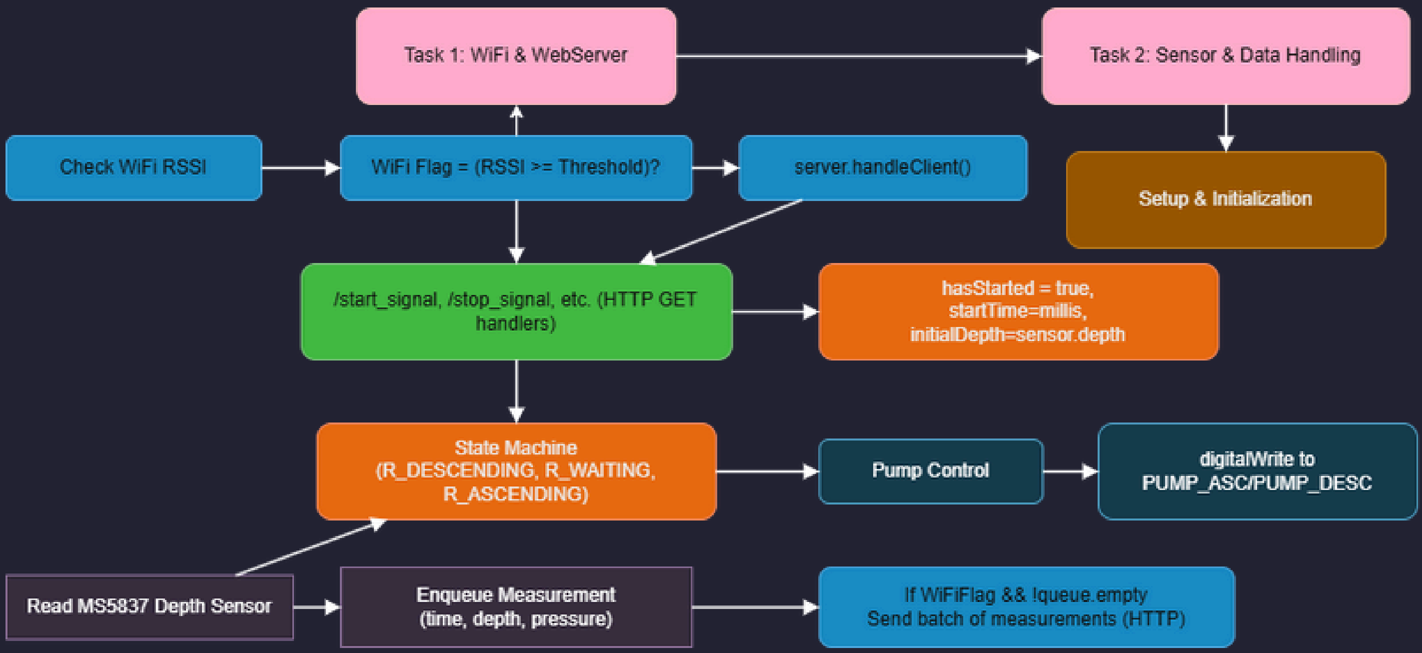

Figure 3: Flow chart of the Multi-Core Logic