

Float description

We designed our float using a 41.5 cm pvc pipe as the main body . The pipe is sealed from both sides using HDPE (Artelon) caps with O-rings .The caps are sealed using a multi-layer insulation system, including a gasket layer, silicone sealant, acrylic sheets and Aluminium sheets.

The system's operation is centered on the precise balance between its total weight and the buoyant force of the displaced water. We achieve neutral buoyancy at the point where these two forces are equal. Once this equilibrium is established, the float's vertical trajectory can be finely adjusted by making incremental changes to the internal water volume.



Figure (1): Model of the float assembly



Figure (2): Clarification of the buoyancy engine mechanism

Buoyancy engine

Our main goal was to reach the neutral buoyancy point when the weight force is equal to the bounce force. So we designed our buoyancy engine consisting of four syringes , each with a capacity of 60 ml. The syringes are mechanically connected to a lead screw mechanism, the lead screw controls the linear motion of the syringe plungers, allowing water to be injected or removed from the system. Neutral buoyancy is achieved when approximately 15 ml of water is present in each syringe.

At the neutral buoyancy point;

Weight force = buoyant force

$$\bullet W = (m_1 + m_2) * g$$

m_1 = Mass of float

m_2 = Mass of water inside the buoyancy chamber controlled by the syringe

$$\bullet F_b = \rho * V_t * g$$

$$\rho = 1000 \text{ kg/m}^3$$

m_t = Total volume (tube + legs)

Electrical System

The electrical system of the float is based on an ESP32 microcontroller which acts as the main controller for the system. It is responsible for controlling the stepper motor driver, reading sensor data. The motor control section uses a stepper motor driver to control the stepper motor through two control signals from the ESP32: **STEP** and **DIR**. The driver is powered by a 12V supply and includes a capacitor to stabilize the voltage. The power system distributes the battery voltage to different parts of the circuit. The battery input passes through protection components such as fuses before powering the system. A 5V buck converter is used to regulate the voltage for components that require a lower supply voltage. The system also includes several sensors. A sensor is connected to the ESP32 to measure depth. Additionally, We utilized a 24-bit ADC to enhance the system's accuracy and precision, since its high resolution enables more precise signal acquisition and significantly improves measurement reliability.

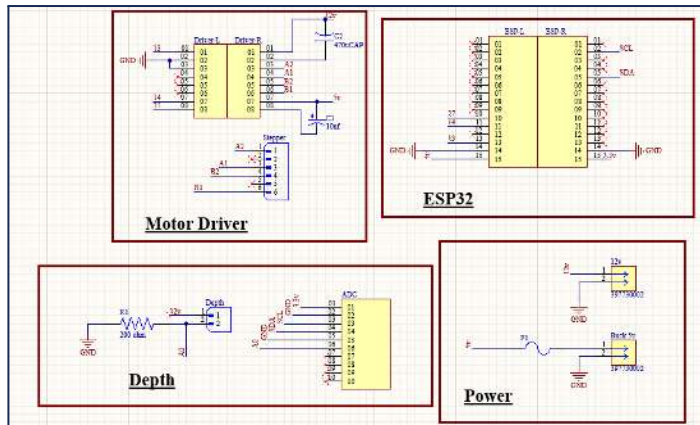


Figure (3): Schematic

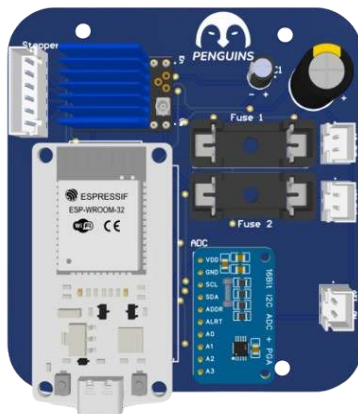


Figure (4): PCB

Battery

The float system is powered by a 12V battery pack **10x AA-1.2V - Ni-Mh Set Of 2500mAh**. The battery output passes through a 2A fuse to protect the system from overcurrent. After that, the power goes to a relay, which acts as a safety control to disconnect the battery in case of a leak detection.

The power is distributed into three paths. The first path supplies 12V directly to the motor driver, which powers and controls the motor. The second path goes to a 12V to 5V buck converter that steps down the voltage to provide a stable 5V supply for the ESP32 microcontroller. The third path is dedicated to supplying the ADC with power from the 12V battery, ensuring stable and reliable operation.

A 1A fuse is placed between the buck converter and the ESP32 to protect the control circuit. The ESP32 then powers and communicates with the depth sensor, while also handling Wi-Fi communication with the external system. This power distribution design ensures that high-power components such as the motor operate safely while sensitive electronics receive stable and protected power.

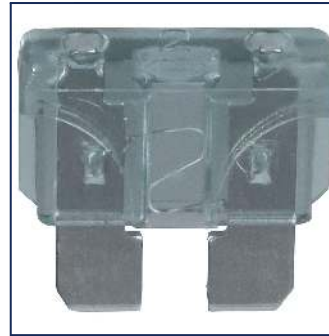


Figure (5): 2A Fuse



Figure (6): Battery Pack

	Current (A)	Voltage (V)	Power (W)
Motor	2 * 0.8 (for one coil) ≈ 1.6	12	$1.6 * 12 = 19.2$
ESP 32	0.3	5	$0.3 * 5 \approx 1.5$
Depth	0.02	12	$0.02 * 12 = 0.24$
Total			20.94

Table (1): Power Calculations

Based on the total current calculation (~ 1.745 A), a **2A fuse** is selected to protect the system.

Wireless Data Transmission

The onboard ESP microcontroller serves as the primary data acquisition node. It polls the pressure sensor, calculates depth, and formats the telemetry (Company ID, Timestamp, Pressure, Depth) into standardized JSON packets. These packets are transmitted to the Topside Node.js server at a minimum rate of one packet every five seconds. To prevent data loss during network drops, the ESP implements a local RAM/Flash queue, actively buffering failed transmissions and batch-sending them upon Wi-Fi reconnection.

GUI Integration & Persistence

Upon ingestion, the Node.js server synchronously appends each valid packet to a persistent .json file, ensuring mission data survives any unexpected system restarts. The server simultaneously broadcasts the telemetry to the React dashboard via WebSockets. The GUI dynamically logs these packets into a real-time verification table featuring conditional auto-highlighting. This logic automatically flags sequential packets falling within the required depth tolerances (Target 1 at 2.5m and Target 2 at 0.4m), instantly verifying the 30-second depth holds for the judges.

Float SID

Fuse calculation

Total power draw:

28.8W

Total current draw:

2.4 A

Full load amperage:

1.33 A

Fuse used:

2 A

12V Line

5V Line

Signal Line

Motor Line

Wifi

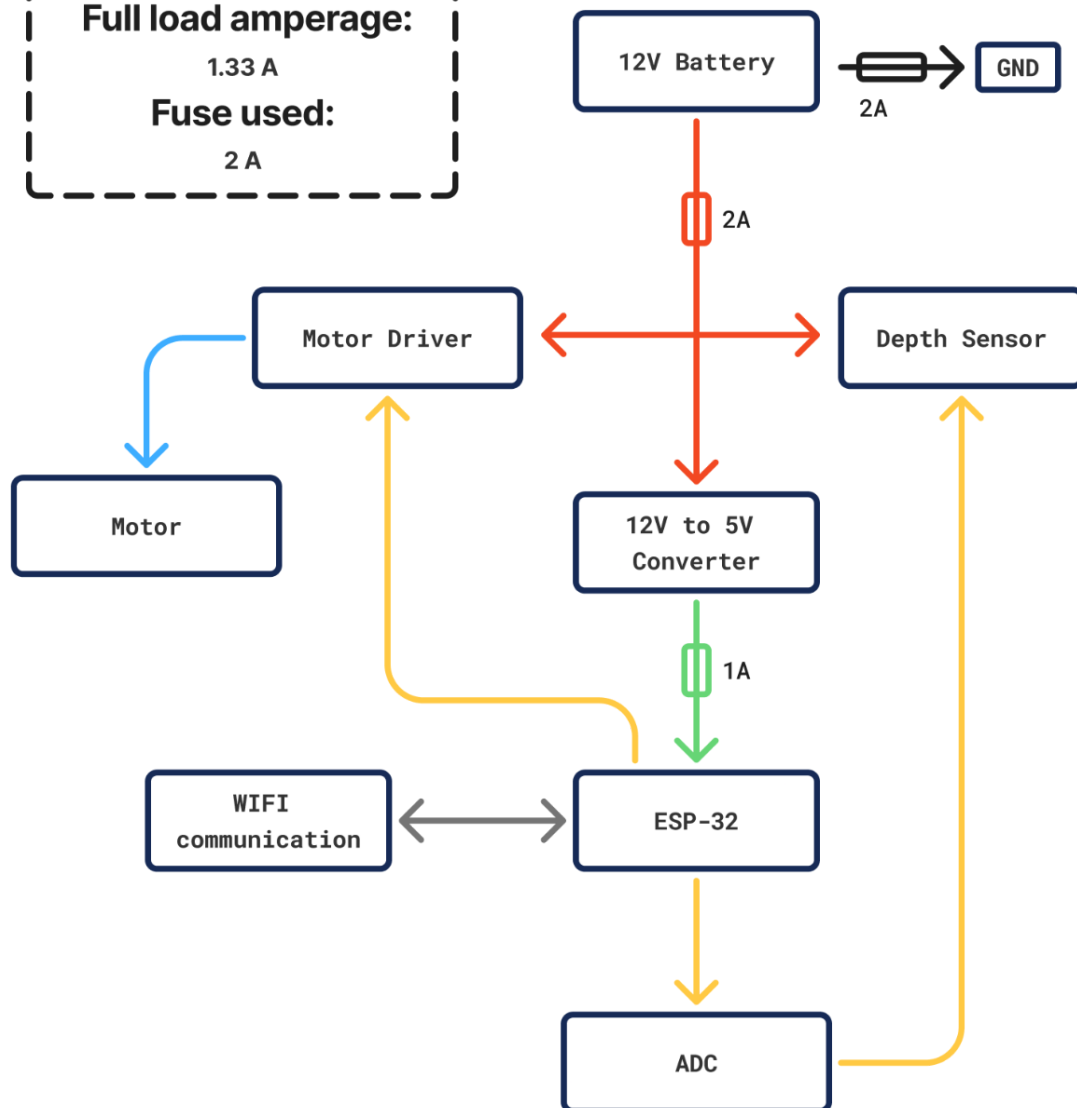


Figure (7): Non-ROV Device SID