

Goal: Implement a cooperative scheduler on an ESP32 to manage multiple tasks within a Remotely Operated Vehicle (ROV) system.

1. Task Identification:

- **Sensor Data Acquisition:** Reading data from the MPU6050 accelerometer and gyro via I2C.
- **Motor Control:** Generating PWM signals to control ESCs for the motors.
- **Ethernet Communication:** Handling data transmission between the ROV and the land unit.
- **User Command Processing:** Executing commands received from the user.
- **Camera Data Streaming:** Managing the video feed from the camera.
- **Communication with Raspberry Pi (SoC):** Exchanging data between the MCU and the Raspberry Pi.
- **Battery Monitoring:** Monitoring the voltage and health of the battery pack.
- **System Monitoring:** Checking the overall health of the system.
- **Battery Pack Management:** Managing charging and discharging of the battery packs.
- **Relay Control:** Controlling relays connected to various actuators.

2. Prioritization Algorithm:

- **Rate Monotonic Scheduling (RMS)** was suggested, where tasks with higher periodicity (i.e., those that need to run more frequently) are given higher priority.

3. Task Priorities:

- Tasks were assigned priorities from 1 to 10, with 1 being the highest and 10 being the lowest.

Task	Priority	Reasoning
Sensor Data Acquisition	1 (Highest)	Critical for stabilization and control; requires frequent updates.
Motor Control	2	Essential for ROV movement; needs quick response to control inputs.
Ethernet Communication	3	Crucial for command processing and data transmission to the land unit.
User Command Processing	4	Important for executing user commands; slightly less critical than control tasks.
Camera Data Streaming	5	Important for navigation; can buffer data if necessary.
Communication with Raspberry Pi	6	Supports other tasks with less frequent data exchange.
Battery Monitoring	7	Ensures the ROV doesn't run out of power; can be checked periodically.
System Monitoring	8	Maintains overall system health; can run periodically.
Battery Pack Management	9	Important for battery health; can be managed in the background.
Relay Control	10 (Lowest)	Likely controls non-essential systems; can run when other tasks are not active.

4. Scheduler Implementation:

- **Fixed-Priority Preemptive Scheduling** was suggested, allowing higher-priority tasks to preempt lower-priority ones.

- **Implement a yielding mechanism** to allow tasks to give up control voluntarily.