Mission: Mars – Emergency Reaction Wheel Redesign

Problem Statement

You are part of a Mars-bound mission aboard the spacecraft **Aether-1**. Midway through the journey, the spacecraft begins to tumble — your primary reaction wheel has failed. With communications lag increasing and ground support limited, it's up to you and your onboard engineering team to save the mission.

Luckily, you have access to a 3D printer, circuit fabrication tools, and embedded computing modules. Your task is to design, power, and control a brand-new reaction wheel system that can stabilize the spacecraft before it drifts off-course permanently.

Be sure to check the required values and constraints given at the end of this document.

Task 1: CAD Design of the Reaction Wheel

Objective

Design a 3D printable reaction wheel suitable for spacecraft attitude correction. The design should:

- Include the motor mount, flywheel, shaft, and housing.
- Be compact, mass-efficient, and mechanically balanced.
- Be printable on a 3D printer using available materials.

Deliverable

- A CAD file (e.g., `.STEP` or `.STL`) of the complete reaction wheel assembly.
- A short description of design decisions.

Task 2: Power and Motor Driver PCB Design (KiCad / Proteus)

Objective

Design a basic power and motor driver circuit to support the operation of a reaction wheel system while keeping the system energy-efficient and modular. The design must:

- Be powered by a typical spacecraft-standard DC input.
- Be energy efficient and support modular design.
- Prioritize clear part placement on a 10 cm × 10 cm PCB base (no full trace routing required).
- Include optional feedback sensors (e.g., A3144 Hall sensor, MPU6050). An IMU such as the MPU9250 is required.

Components

- **Motor:** BLDC motor or brushed DC motor.
- **Motor Driver IC:** DRV8833, L298N, or ESC (for BLDC).
- **Voltage Regulator:** LM2596, LM317.
- **Passive Components:** Resistors, capacitors, and diodes for noise filtering, decoupling, and protection.
- **Sensors (Optional):**
  - Hall Sensor: A3144
  - IMU: MPU6050 (optional), MPU9250 (required)
  - Thermistor for temperature monitoring (optional)

Software Tools

- KiCad (preferred, open-source)
- Proteus
- EasyEDA (alternative)

Deliverables

- **Circuit Schematic (.sch):** Showing all functional blocks (power regulation, motor driver, sensor connections, etc.) along with passive components.
- **Partial PCB Layout (.kicad_pcb / .dsn):** Realistic 10cm × 10cm board footprint with logical and space-efficient placement of components.
- **Explanation PDF:** A short document including:
  - Description of voltage flow and overall power delivery logic.
  - Justification for choice of components and layout arrangement.
  - Strategies used to ensure low power consumption and thermal stability in a space-like environment.

Task 3: Control Algorithm Development for Attitude Stabilization

In this task, you will implement the control logic for your reaction wheel system to achieve orientation control in a zero-gravity environment. The controller must process sensor feedback and generate appropriate control signals to stabilize the system using the reaction wheel.

Functional Goals

- Use sensor feedback (e.g., gyroscope or IMU) to measure angular position or velocity.
- Implement a closed-loop control strategy to bring the system to a desired orientation or maintain stability against disturbances.
- The control signal should drive the reaction wheel motor via appropriate driver circuitry based on your schematic (Task 2).
- Your code must run on the microcontroller you've selected and integrate with the designed hardware.

Hint: Think in terms of feedback control systems where the system continuously adjusts its output based on the difference between desired and actual behavior.

Deliverables

- A complete '.ino' file (Arduino sketch) implementing your control algorithm.
- The sketch should include:
  - Initialization of all relevant sensors and actuators.
  - Sensor reading logic.
  - Your control algorithm.
  - Output logic (PWM or digital control signals to motor driver).
- Inline comments explaining key logic and tuning parameters.
- A short README or comment block at the top of the code describing:
  - The microcontroller used.
  - Sensor(s) and actuator(s) used.
  - How the control system behaves under different test conditions.

Resources

Task 1: CAD Design

Software: SolidWorks, Fusion 360, or FreeCAD
Tools: MATLAB (for inertia, torque, and flywheel calculations)

Task 2: Power System Design

Link

Task 3: Control System (PID)

- **Introduction to PID Controllers:** Proportional–Integral–Derivative (PID) Controller – GeeksforGeek
- **Basics on ESP32 Microcontroller and Arduino Programming:**
  Getting Started with Arduino on the ESP32 – Espressif Docs
  Arduino Official Tutorials – Arduino Docs

Satellite Data and Constraints

Parameter	**Value**
Max mass of satellite	3 kg
Internal dimensions	10 cm $\times$ 10 cm $\times$ 22 cm
Moment of inertia (I)	0.0195 kg·m^2 (longitudinal axis)
Initial angular velocity	5°/s ($\approx$ 0.087 rad/s)
Target angular velocity	$\leq$ 0.5°/s ($\approx$ 0.0087 rad/s)
Detumbling time window	120–300 seconds
Volume allowed for reaction wheel assembly	$\leq$ 250 cm^3
Centre of mass	At geometric center

Note:

- This is a team event. Teams of 2–4 students may register.
- Ideally, your team should include members with diverse skill sets — such as CAD modeling (e.g., SolidWorks), electronics/circuit design (e.g., KiCad or Proteus), and control system programming (e.g., PID logic in Arduino/Python).
- Don't worry if you're not proficient in all areas — this is a great opportunity to learn! Collaborate, divide tasks smartly, and grow together.
- You are encouraged to submit whatever components you are able to complete — even if it's just the CAD file or only the control code.
- This is a learning-centric challenge — creativity, effort, and clarity are valued just as much as completeness.
- Use this chance to explore interdisciplinary skills and practice real mission-critical engineering design!