

Embedded Hardware Designer Challenge: Sensor Interface & Debug

Time Limit: 1.5–2 hours

Your Mission

Design a temperature-sensing system, debug a mysterious UART failure, and show us how you think.

(Note: No soldering required—just creativity, logic, and a dash of caffeine.)

Task 1: Design the System (45–60 mins)

Objective: Create a microcontroller-based circuit to read temperature data and transmit it via UART.

1. Schematic Design (30 mins)

- Sketch a schematic (tool-based or hand-drawn) for:
 - An STM32 (or similar) microcontroller interfacing with an I2C temperature sensor (e.g., TMP102).
 - LED blinking at 1 Hz during data sampling.
 - UART interface (e.g., USB-to-serial) for communication at 115200 baud.
- **Bonus:** Note 1–2 key signal integrity or manufacturability considerations.

2. Firmware Logic (30 mins)

- Write pseudocode or C snippets to:
 - Initialize I2C and UART.
 - Read sensor data periodically and blink the LED.
 - Transmit temperature readings.
 - **Bonus:** Briefly explain power-saving strategies for a battery-operated device.
-

Task 2: Debug the Chaos (30–45 mins)

Problem: The LED blinks, but the UART output is gibberish.

1. Debugging Plan (15 mins)

- Outline steps to diagnose the issue (tools, signal checks).
- Hypothesize 2–3 root causes (e.g., baud rate mismatch, faulty pull-ups and how their symptoms would present).

2. Collaboration (15 mins)

- How would you explain the issue to a software engineer?
 - What common mistake might a junior engineer make in this design?
-

Evaluation Criteria

- **Technical Rigor:** Schematic completeness/elegance, firmware logic & tidiness, debugging methodology.
 - **Problem-Solving:** Efficiency in isolating issues, creative solutions.
 - **Communication:** Clear explanations and approach strategy, teamwork mindset.
 - **Bonus:** Mention of power optimization, EMC/EMI, or manufacturability.
-

Why You'll Enjoy This

- **Hardware Geeks:** You'll design circuits where every trace and component counts.
 - **Firmware Wizards:** You'll write code that bridges hardware and software seamlessly.
 - **Team Players:** Imagine this as a tiny preview of collaborating with a team that's as passionate about details as you are.
-

Submit: Your schematic, code snippets, debugging notes, and a **brief note** on what excites you most about embedded systems work.