

UW MedTech Resolve - Software Onboarding Task

Welcome to the team! Your onboarding task is to design and implement a sensor simulation system using **C or C++**. This system is open-ended by design — we encourage you to apply your creativity in architecture, implementation style, and extensibility.

The primary goal is to demonstrate your skills in multithreading, file handling, inter-process communication, and robust error handling, while delivering clean and maintainable code.

System Overview:

Your system should be composed of the following components:

- **Sensor Device Simulator:** generates periodic temperature readings and stores them in a shared buffer
- **Data Logger:** collects the sensor readings and writes them to a log file
- **Command Interface:** allows external interaction via a Unix socket
- **Build & Test Automation Script:** bash script to compile, run, and test the system

Module Requirements:

1. **Sensor Device Simulator:**
 - Runs in a separate thread and generates random temperature readings (e.g., between -50°C and 150°C) every 500 milliseconds.
 - The generated data should be stored in a shared buffer (e.g., a queue protected by a mutex).
2. **Data Logger:**
 - Runs in another separate thread and periodically writes sensor data to a log file "sensor_log.txt" every 2 seconds.
 - It should append new readings without overwriting previous data.
3. **Command Interface via Unix Socket:**
 - Listens on a socket "/tmp/sensor_socket"
 - Accepts the following commands:
 - GET: Returns the last five temperature readings
 - STOP: Stops the simulation gracefully
4. **Bash Script for Automation:**
 - Compiles the program
 - Runs the test scenarios and summarize the results

Bonus Features (Optional):

- Use JSON format for the output of the sensor logger
- Implement a timeout for the socket to avoid blocking indefinitely
- Extend the sensor simulation to support multiple sensor types (e.g., humidity, pressure) via runtime configuration
- Anything else that you would find useful

Deliverables:

Candidates must submit:

- C/C++ source code implementing the sensor simulation system
- Unit tests covering key functionalities
- The main bash script to trigger the demonstration
- A short documentation (could be a README) to explain how to build, run, and test the code