CS-499-M01 Computer Science Capstone

Professor: Pravin Bhandari

Maria Duhamel

August 15, 2025

The artifact I selected for my ePortfolio is titled “Wiring an LED Display,” created during CS 350: Emerging Systems Architectures and Technologies in Term 2025 C-3 (May–June). It is a thermostat system that integrates sensors, LEDs, an LCD, and UART communication to simulate cloud data transmission.

I chose this project because it demonstrates my ability to design and implement a complex embedded system that integrates both hardware and software components. The thermostat system I developed uses sensors, an LCD, LED indicators, and UART communication to simulate cloud data transmission. I also evaluated three hardware platforms, Raspberry Pi, Microchip, and Freescale, to determine the most suitable option for real-world deployment. This artifact showcases my skills in system analysis, peripheral integration, and hardware selection, all of which are essential in embedded systems development.

Another reason I selected this project is that I genuinely enjoyed working on it during that term. It was challenging but incredibly rewarding. Exploring and figuring out how to make something work truly excites me.

Regarding course outcomes, I believe I have met my planned goals related to embedded system design and hardware-software integration. At this time, I do not need to update my outcome-coverage plans.

In enhancing this artifact for the capstone, I focused on applying advanced software engineering practices to improve its functionality and maintainability. This process allowed me to deepen my understanding of modular software design, real-time data handling, and embedded system integration. One of the challenges I faced was ensuring the smooth temperature data remained responsive enough for real-time control, which required tuning the size of the moving average window.

The enhancements I incorporated for Milestone Three specifically relate to algorithms and data structures. I implemented a moving average algorithm using collections.deque to smooth out temperature readings and reduce sensor noise and fluctuations. I also replaced all calls with the original getFahrenheit() function with a new get\_smoothed\_fahrenheit() function and updated the logic in updateLights(), manageMyDisplay(), and setupSerialOutput() to use the smoothed temperature values. This demonstrates my ability to apply efficient data structures and implement real-time data processing algorithms in an embedded system context.

In addition to algorithmic improvements, I integrated an SQLite database to log temperature readings, system states, and set points over time. I designed the database schema with constraints to ensure data integrity and added indexes to optimize query performance. I also created a SQL view to summarize average temperatures by system state, enabling quick analysis of system behavior. These enhancements demonstrate my ability to design and implement efficient, quarriable data storage solutions within a resource-constrained embedded environment. This not only improves the system’s reliability and traceability but also aligns with real-world practices in IoT and edge computing, where persistent data logging and analysis are essential.

These changes are beneficial because they improve both the reliability and responsiveness of the thermostat system. By smoothing out temperature readings, the system avoids reacting to sudden spikes or drops caused by sensor noise, resulting in more stable and accurate control. Additionally, the integration of an SQLite database adds a critical layer of data persistence and traceability. With structured logging of temperature, state and set point data, the system now supports historical analysis and performance monitoring. These enhancements not only improve the user experience but also reflect real-world practices in embedded and IoT systems, where data integrity, efficient processing, and long-term storage are essential for robust and scalable solutions.