Jaden Williams

SNHU – CS499

7/22/25

Milestone 3

The artifact is a Python-based thermostat control system developed for a Raspberry Pi, originally created during CS350: Embedded Systems. The project was developed in Spring 2025 and is designed to read real-time temperature and humidity data from sensors, control a heating or cooling mechanism using GPIO pins, and display status information on LCD. It also supports user input via buttons to cycle thermostat modes and adjust temperature setpoints.

I selected this artifact for my ePortfolio because it demonstrates a solid application of embedded systems programming, sensor integration, GPIO control, and user interaction on a microcomputer platform. These are real-world, applied software engineering skills that reflect a professional-grade understanding of hardware-software interfacing and control systems design.

This artifact shows my ability to:

* Design a state machine architecture for hardware control
* Interact with physical sensors (DHT11) and display units (LCD)
* Modularize and document Python code for readability and future extensibility
* Implement user input and feedback loops via GPIO buttons and serial output

The artifact was further improved by refactoring it for modular design, improving commenting and documentation, and preparing the codebase for future enhancements, including misting control and database integration.

I met my planned course outcomes related to designing and evaluating computing solutions using algorithmic principles. I demonstrated the use of well-structured classes, effective control flow, and real-time logic. This artifact now aligns more directly with outcomes such as:

* Designing and evaluating computing solutions using algorithmic principles and computer science practices.
* Implementing computer solutions using well-founded and innovative techniques and tools.

No major updates to my outcome coverage plan are required, but this section of the project more clearly aligns with those particular outcomes now that the control logic and GPIO-based user interface have been added.

The enhancement process challenged me to think like an embedded systems developer, balancing algorithm design with hardware limitations and physical interaction. One of the biggest learning moments came while handling GPIO input to safely and accurately adjust setpoints—specially making sure that accidental or bouncing inputs didn’t cause unexpected behavior. I also had to ensure the system was modular enough to be extended in future phases, such as adding persistent storage or networking.

This exercise pushed me to reflect not just on writing code that works, but on writing code that is maintainable, testable, and responsive in real-time environments. It deepened my appreciation for the nuances of control systems and highlighted the importance of clean abstractions and robust logic when working with hardware.