Category One Enhancement Narrative

Briefly describe the artifact. What is it? When was it created?

The artifact in use is the final project from the emerging systems architecture and technologies course taken last semester. This project was created as an example of the embedded systems knowledge gained. The code in the project was intended to create and prototype an embedded system smart thermostat. Through the use of peripherals, a current temperature would be sensed, compared with a setpoint temperature, and if necessary, the LED would light (indicating a heating system turning on). Two buttons on the TI board were used to manipulate the setpoint temperature.

Justify the inclusion of the artifact in your ePortfolio.

This artifact was selected for inclusion in the ePortfolio as experience with programming for embedded systems appeared to be a self-promoting skillset, especially with pertinence to IoT devices.

Within the artifact, improvements to the code's efficiency through exchanging the current conditional statements for a task scheduler as well as generally altering the code to conform to code review improvements will showcase an ability to deliver a product that accomplishes industry specific goals (such as a smart thermostat that simulates data being sent to a server via WIFI). Initially, the intent to prototype a more complete air system that simulates AC capabilities in addition to heat would have lent to this idea as well (see challenges below).

Overall improvements to the code also include cleaning up unnecessary and unused variables and remaining test stubs/leftover code attempts to ensure proper use of memory. Consistency and

sufficiency of documentation and formatting was ensured to aid in maintainability and readability. Lastly, the issues surrounding the output display of data were addressed.

Did you meet the course objectives you planned to meet with this enhancement in Module One? Do you have any updates to your outcome-coverage plans?

While the considerations of WIFI solutions being weighed, involving managing tradeoffs between capabilities, peripheral support, and sufficient memory for code execution, pertains to the course outcome of managing trade-offs involved in design choice, that was not the main outcome intended to be met.

The main objective of demonstrating an ability to use techniques and tools to implement solutions that deliver value and accomplish industry specific goals is met through the use of embedded systems hardware and development environments, peripherals and tools, and coding techniques (ex. task schedulers).

Reflect on the process of enhancing and modifying the artifact. What did you learn as you were creating it and improving it? What challenges did you face?

Pin configuration: In attempting to make the main functional enhancement, the AC counterpart to the current heating system prototype, I was able to learn more about the hardware itself. In configuring the hardware and pins for peripheral objects, I found that you are rather limited by the pins available on the board. Multiple LEDs have been used successfully in previous work: using pulse width modulation (PWM) to create morse code messages using two LEDs and using GPIO with two LEDs. In the former PWM example, the LEDs were configured using the I2C pin. In the latter GPIO example, the red LED has a pin available for selection in pin configuration (P64/29). Neither the yellow nor green LEDs have a similar option under GPIO pin configuration (although both have a hardware option). In order to use a second LED with the

red LED in the past using GPIO, the second LED pin was configured as the I2C pin (P02/10) as well. In the previous projects the I2C peripheral was not in use, whereas in this project it is integral. The workarounds I was able to research involved more hardware, such as a GPIO expander or PWM controller. A seemingly simple approach to configuring a logically similar but opposite AC counterpart to the heat system functionality would require additional hardware and hardware integration that would be outside of the scope of this software category.

Task Scheduler: As part of the enhancements for this project, I implemented the use of a task scheduler to increase efficiency. In order to accomplish this, I created a structure to hold relevant attributes (such as task period, elapsed time, and task function) of a task (button checking, comparing temperatures, and outputting data) in the program. Additionally, a task initializer function and a task scheduler function were created to populate the tasks and their attributes, and to handle the main task scheduler for loop. This involved moving repetitive local variables throughout the code to global variable scope and decreasing the overall amount of declared variables. This also involved creating individual task functions with respective functionality. One issue encountered was when calling the task functions in the task scheduler function originally, I got an error (“called object is not a function or function pointer”). After researching and experimenting around the error, I defined a function type and used this type in the overall task structure to resolve the issue. Ultimately this implementation increased the efficiency of the code, while decreasing the overall amount of repetitive code in the main function.