



Competencies

In this project, you will demonstrate your mastery of the following competencies:

- Write interface software to control hardware components
- Analyze fundamental considerations regarding hardware architecture design and their impact on application performance
- Recommend and justify emerging systems architectures and technologies based on business requirements

Scenario

The global smart thermostat market is forecast to reach almost \$9 billion by 2026. The CEO of SysTec wants to get into this lucrative market and has tasked your engineering department with creating a smart thermostat using the TI board. SysTec develops analytics software for servers, but the CEO remembers from your interview that you took an embedded systems course and asks you to build a prototype. The final goal is to develop a thermostat that sends data to SysTec's server software over Wi-Fi, but first they need a prototype of the low-level thermostat functionality working.

For the prototype, you will use the TMP006 temperature sensor to read the room temperature (via I2C), an LED to indicate the output to the thermostat where LED on = heat on (via GPIO), two buttons to increase and decrease the set temperature (via GPIO interrupt), and the UART to simulate the data being sent to the server.

You will also create a written report for your team to ensure that the system you created is based on SysTec's business requirements and technical specifications. While your team is testing your code, you have been asked to architect the next phase of the project in your report: connecting the thermostat to the cloud. Although you prototyped on the TI development board, other manufacturers make integrated Wi-Fi solutions. In this next phase, you are going to analyze various hardware architectures (from TI, Microchip, and Freescale) and recommend and justify the architecture decision based on the following business requirements:

- The thermostat must support the peripherals used in the project.
- The thermostat must connect to the cloud via Wi-Fi.
- The architecture chosen must have enough Flash and RAM to support the code.

In your report, you will be responsible for comparing and contrasting the three hardware architectures based on the above metrics.

Directions

Begin your work by accessing the [Project Thermostat Lab Guide PDF](#) document. While this document was written for a Windows interface, the tools can be used on Mac or Linux as well. To accomplish the work outlined in the guide, you will need the following:

- TI CC3220x LAUNCHXL
- TI Code Composer Studio (installed)
- USB connection between the PC and board

Specifically, you must address the following rubric criteria:

1. Develop **code** for all of the specified **functionality**. Remember, the goals of this project include reading room temperature, indicating the result via LED output, increasing or decreasing the set temperature, and simulating the data being sent to a server.
2. Create code that initializes the **timer** and uses it to drive specified actions. This involves two steps, both the initializing of the timer and then its use in running an action.
3. Create code that uses **interrupt** to detect button presses. This involves two steps, both the detection of the button presses and then using the result to run an action.
4. Create code to initialize the **I2C peripheral** and use it to read the temperature sensor. This code must both initialize the I2C peripheral and then read the temperature sensor correctly.
5. Create code to initialize the **GPIO peripheral** and use it. Remember that the GPIO is involved in indicating the output of the temperature via LED and setting the temperature with two different buttons (one to increase and one to decrease temperature).

6. Create code to initialize the **UART peripheral** and output specified data. The UART should be used to simulate data being sent to the server. Be careful to ensure that the UART is initialized to the correct baud rate and serial configuration.
7. Implement (in code) the **task scheduler functionality**. This should match the specifications described by the Project Thermostat Lab Guide.
8. Provide the appropriate technical and operational **documentation** for the **task scheduler** that you created to satisfy step 7. This should address all the required functionality and be documented according to industry standards describing the algorithm used, its inputs and outputs, and expected results. To excel for this item, you may additionally provide an appropriate state-machine diagram in addition to the other documentation.
9. Apply coding **best practices** in formatting, commenting, and functional logic.
10. Explain how the thermostat supports the **peripherals** used in the project. Make sure that you have included all the required details from the scenario in your report. You should discuss each of the three outlined hardware architectures, including TI, Microchip, and Freescale.
11. Explain how the thermostat connects to the **cloud** via Wi-Fi. Discuss all three architectures in your work.
12. Discuss the architecture's **Flash and RAM** that supports the code. Include all three architectures in your discussion.

What to Submit

To complete this project, you must submit the following:

Task Scheduler

The task scheduler should be created as a draw.io file and then exported as a PDF file for submission in Brightspace.

Code

Zip your workspace with all of your code and then submit the zipped file for grading.

Video

Submit a video demonstration of your project's functionality. While your code is running, record your board and narrate to describe what the LEDs are doing. Be sure that you push your button during the video to show what happens. Your video should not be longer than one minute. If you encounter any difficulties filming the lights on your hardware component, please reach out to your instructor.

Report

Your reflections should be documented in a 1- to 2-page Microsoft Word document with 12-point Times New Roman font, double spacing, and one-inch margins. Sources should be cited according to APA style.

Project Rubric

Criteria	Exemplary (100%)	Proficient (85%)	Needs Improvement (55%)	Not Evident (0%)	Value
Code Functionality	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner	Develops code for all of the specified functionality	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include developing for the individual requirements and ensuring that they work together as intended	Does not attempt criterion	10
Timer	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner	Creates code that initializes the timer and uses it to drive specified actions	Shows progress toward proficiency, but with errors or omissions; areas for improvement may	Does not attempt criterion	8

			include initializing the timer and then also using it to run an appropriate action		
Interrupt	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner	Creates code that uses interrupt to detect button presses	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include detecting the button presses and then using the result to run an appropriate action	Does not attempt criterion	8
I2C Peripheral	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner	Creates code to initialize the I2C peripheral and use it to read the temperature sensor	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include both initializing the I2C peripheral and then using it to run an appropriate action	Does not attempt criterion	8
GPIO Peripheral	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner	Creates code to initialize the GPIO peripheral and use it	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include both initializing the GPIO peripheral and then using it to run an appropriate action	Does not attempt criterion	8
UART Peripheral	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner	Creates code to initialize the UART peripheral and output specified data	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include both initializing the UART peripheral and then outputting appropriate data	Does not attempt criterion	8
Task Scheduler Functionality	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner	Implements the task scheduler functionality	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include implementing all intended functionality and matching the specifications to the provided description	Does not attempt criterion	10

Task Scheduler Documentation	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner, or provides an appropriate state-machine diagram in addition to meeting proficiency in their documentation	Describes the operation and technical specification through task scheduler documentation	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include addressing all required functionality and documenting it according to industry standards	Does not attempt criterion	10
Best Practices	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner	Applies coding best practices in formatting, commenting, and functional logic	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include use of formatting best practices to make code easy to read, commenting best practices to ensure code is clearly explained, or functional logic so the program runs as expected	Does not attempt criterion	5
Report: Peripherals	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner	Explains how the thermostat supports the peripherals used in the project	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include discussing all three architectures or using evidence to support the claims	Does not attempt criterion	6
Report: Cloud	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner	Explains how the thermostat connects to the cloud via Wi-Fi	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include discussing all three architectures or using evidence to support the claims	Does not attempt criterion	6
Report: Flash & RAM	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner	Discusses the architecture's Flash and RAM that supports the code	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include discussing all three architectures or using evidence to support the claims	Does not attempt criterion	6
Articulation of Response	Exceeds proficiency in an exceptionally	Clearly conveys meaning with correct	Shows progress toward proficiency,	Submission has critical errors in	3

	clear, insightful, sophisticated, or creative manner	grammar, sentence structure, and spelling, demonstrating an understanding of audience and purpose	but with errors in grammar, sentence structure, and spelling, negatively impacting readability	grammar, sentence structure, and spelling, preventing understanding of ideas	
Citations and Attributions	Uses citations for ideas requiring attribution, with few or no minor errors	Uses citations for ideas requiring attribution, with consistent minor errors	Uses citations for ideas requiring attribution, with major errors	Does not use citations for ideas requiring attribution	4
Total:					100%