Southern New Hampshire University

CS 350 Final Project I Guidelines and Rubric

Overview

Competent software developers possess not only the knowledge and skills to code proficiently but also a good understanding of computer architecture. Why is this important? As new technologies emerge, companies have opportunities to develop products and services that capitalize on these emerging technologies. Given the current tools, architectures, and tech, think about the opportunities you see in what is emerging. How can you create solutions for an existing problem or new innovation? An engineering mindset is an important dispositional trait you can cultivate to leverage these opportunities. You may write not only desktop applications but also software to control hardware and software components. You may also write software that provides display output from a device.

The final project in this course is meant to help you develop your engineering mindset. The submission for this final project is in two parts: **Final Project I** is the creation of scripts that run an embedded system device. For this part, read the scenario in the prompt section of this document, which details your role at a company called SysTec. In **Final Project II**, you will be immersed in a different scenario to develop a response to a request for information document from Arboretum Research Centers. Using the updated specifications you will be provided with in that scenario, you will recommend suitable embedded systems and software architectures necessary to meet those specifications.

In this assignment, you will demonstrate your mastery of the following course outcomes:

- CS-350-01: Write software to interface with embedded systems to control hardware and software components
- CS-350-02: Display data from an embedded device by modifying interfacing software in an emerging systems architecture

Prompt

Final Project I will build on the milestone assignments you completed in Modules Two through Five. This work creates a foundation that you will then modify for the client needs detailed below, which you will address in Modules Six and Seven of the course. You will extend the capability of the weather station prototype and build the custom software components that control the sensors and the sending, receiving, and storage of data.

Scenario:

You are currently working for an embedded systems development and engineering company called SysTec; you are working in the role of systems engineer, and you are responsible for product development, embedded systems, and prototype coordination. SysTec develops a flagship product that can be used to monitor changing weather conditions in remote environments. This product, called WetSpec, is the perfect delivery system for an embedded solution that allows clients to extend the capability of the prototype as much as needed to fit the scope and goals of their current projects, which are mainly related to research and development. Since academic partners are one of the largest clients that SysTec works with, you will be interfacing with a remote team of engineers who will supply changes in scope and development needs to you throughout the course. Your first milestone assignment involves becoming familiar with WetSpec and

setting up your own working prototype so you can communicate effectively about various components and extending capabilities with in-house engineering teams as various needs change.

Your final project involves creating a weather reporting system that gathers and stores information for future usage. The client has given the following specifications:

Light Sensor: Their proposed platform monitors temperature and humidity, but only during daytime hours and during daytime conditions; sensor readings must not be recorded outside of these times or conditions, or we might risk skewing the data the platform collects.

Frequency of Data: During operating hours, readings must be taken once every 30 minutes, and the readings must be incrementally stored in a JSON file. To clarify, each reading must be added to a single JSON file so that a flat file of readings is stored over time.

Output Visual Using LEDs: The new platform must provide a visual indication of the type of data it is recording, as follows:

- Green LED lights up when the last conditions are: temperature > 60 and < 85, and humidity is < 80%
- Blue LED lights up when the last conditions are: temperature > 85 and < 95, and humidity is < 80%
- Red LED lights up when the last conditions are: temperature > 95
- Green and Blue LED light up when the last conditions are: humidity > 80%

Specifically, the following **critical elements** must be addressed. Most of the critical elements align with a particular course outcome (shown in brackets).

I. **Weather Station:** In this part of your final project, you will be assessed on how well you enhanced the weather station prototype.
   A. **Embedded Systems Code** (State Architecture, Functions, Imports, and Variables)
      i. **Interrupts** are functional and aligned with product specifications. [CS-350-01]
      ii. **Frequency** of data gathering has been implemented, is functional, and is aligned with product specifications. [CS-350-01]
      iii. **Embedded systems code** runs without errors, is logically organized, and is properly commented. [CS-350-01]

   B. **Dashboard Controls**, Functionality, and Reporting Feature**:** In this part of your final project, your instructor will examine your code to determine how well you set up and enhanced the weather station prototype so that people can easily see the information that is being gathered by the sensors.
      i. Writes code that **displays data** to the dashboard. [CS-350-02]
      ii. Includes **screenshot** or picture of the weather station data displaying on the dashboard. [CS-350-02]
      iii. **Dashboard code** runs without errors, is logically organized, and is properly commented. [CS-350-02]

Southern New Hampshire University

Milestones

Milestone One: *Adding an LED and Sound Sensor*
In **Module Two**, you will begin the first part of your prototype embedded system by adding a sound sensor that can be used to listen for certain events or sounds. **This milestone will be graded with the Milestone One Rubric.**

Milestone Two: *Displaying Temperature and Humidity With Database*
In **Module Three**, you will use the Grove DHT (digital humidity and temperature) sensor, store the output in a JSON database file, and display it on the Grove RGB LCD display. **This milestone will be graded with the Milestone Two Rubric.**

Milestone Three: *Interrupts Detecting Light Conditions*
In **Module Four**, you will set up the conditions to turn on an LED of your choice when it becomes dark. **This milestone will be graded with the Milestone Three Rubric.**

Milestone Four: *Displaying the Data on a Dashboard*
In **Module Five**, you will display the humidity and temperature data stored in the JSON file in Milestone Two on a web dashboard. **This milestone will be graded with the Milestone Four Rubric.**

Final Submission: Embedded System Device Script(s)
In **Module Seven**, you will submit Final Project I. It should be a complete, polished artifact containing **all** of the critical elements of the final product. It should reflect the incorporation of feedback gained throughout the course. **This submission will be graded with the Final Project I Rubric.**

Deliverables

Milestone	Deliverable	Module Due	Grading
One	Adding an LED and Sound Sensor	Two	Graded separately; Milestone One Rubric
Two	Displaying Temperature and Humidity With Database	Three	Graded separately; Milestone Two Rubric
Three	Interrupts Detecting Light Conditions	Four	Graded separately; Milestone Three Rubric
Four	Displaying the Data on a Dashboard	Five	Graded separately; Milestone Four Rubric
	Final Submission: Embedded System Device Script(s)	Seven	Graded separately; Final Project I Rubric

Southern New Hampshire University

Final Project I Rubric

Guidelines for Submission: Your script(s) and the screenshot of your dashboard must submitted in a single ZIP file. (You can also include a separate dashboard code file in the ZIP, for clarity, if you prefer.)

Critical Elements	Exemplary	Proficient	Needs Improvement	Not Evident	Value
Weather Station: Embedded Systems Code: Interrupts [CS-350-01]	Meets "Proficient" criteria and demonstrates a sophisticated approach to creating interrupts (100%)	Writes code that starts or stops conditions based on client specifications (85%)	Writes code that starts or stops conditions based on client specifications, but with errors (55%)	Does not write code that starts or stops conditions (0%)	16.67
Weather Station: Embedded Systems Code: Frequency [CS-350-01]	Meets "Proficient" criteria and demonstrates a sophisticated approach to coding for frequency of data collection and storage (100%)	Writes code that collects and stores data based on client specifications (85%)	Writes code that collects and stores data based on client specifications, but with errors (55%)	Does not write code that collects and stores data (0%)	16.67
Weather Station: Embedded Systems Code: Embedded Systems Code [CS-350-01]	Meets "Proficient" criteria and demonstrates professional, high-quality embedded systems code (100%)	Writes embedded systems code that runs without errors, is logically organized, and is properly commented (85%)	Writes embedded systems code that runs, is logically organized, and is commented, but with errors or insufficient commenting (55%)	Does not write embedded systems code (0%)	16.67
Weather Station: Dashboard Controls: Displays Data [CS-350-02]	Meets "Proficient" criteria and demonstrates professional, high-quality code (100%)	Writes code that displays data on the dashboard (85%)	Writes code that displays data on the dashboard but with errors (55%)	Does not write code that displays data on the dashboard (0%)	16.67
Weather Station: Dashboard Controls: Screenshot [CS-350-02]		Includes screenshot or picture of data displayed on the dashboard (100%)	Includes screenshot or picture of data displayed on the dashboard, but screenshot or picture shows observable errors (55%)	Does not include a screenshot or picture of data displayed on the dashboard (0%)	16.65
Weather Station: Dashboard Controls: Dashboard Code [CS-350-02]	Meets "Proficient" criteria and demonstrates a sophisticated approach to dashboard code (100%)	Writes dashboard code that runs without errors, is logically organized, and is properly commented (85%)	Writes dashboard code that runs, is logically organized, and is commented, but with errors or insufficient commenting (55%)	Does not write dashboard code (0%)	16.67
				Total	**100%**