Reece Ebel

CS 350

**Reflection on the Development of the SysTec Smart Thermostat Prototype**

**Introduction**

Developing the SysTec Smart Thermostat Prototype showed an opportunity to integrate embedded systems knowledge with a real-world application. The project required designing and implementing a low-level thermostat that reads room temperature using an I2C-connected sensor, controls an LED indicator via GPIO, and simulates data transmission using UART. This reflection will discuss the development process, the challenges that I faced, and decisions made throughout the project.

**Development Process**

The project began by gaining a thorough understanding of the hardware components and their roles within the system. The LED and buttons are controlled using GPIO, with interrupts managing the buttons for responsive temperature adjustments. UART was used to simulate server communication, reflecting how real-world data could be transmitted.

Each peripheral was initialized and configured using TI’s drivers, giving efficient communication and control. The task scheduler was implemented to manage periodic temperature checks, LED status updates, and button press handlings. The scheduler design was important for maintaining a responsive system, which aligns with SysTec’s business requirements for a smart thermostat.

**Challenges and Solutions**

One of the biggest challenges was getting accurate temperature readings and timely responses to user inputs. This required fine-tuning the task scheduler to balance the demands of checking sensor data, updating the LED status, and processing button presses. Another challenge was simulating the data transmission over UART, which needed to be representative of actual server communication without the complexities of implementing a full network stack.

To fix these problems, I used a more modular coding approach, so that each function was responsible for a specific task, such as reading the temperature or controlling the LED. This modular style made it easier to test and debug the individual components, which leads to a more stable system.

**Reflection and Learning**

This project reinforced the importance of understanding both hardware and software aspects of embedded development. The process of integrating all the various peripherals and ensuring they function harmoniously within the system provided valuable learnings into embedded design. Working with interrupts, and task scheduling, showed the complexities involved in creating responsive and reliable embedded applications.

This project truly showed importance of planning and foresight in system architecture. In selecting hardware components that not only met current project requirements but also supported future expansion (like cloud connectivity), I ensured that the system could evolve to meet SysTec’s long-term goals.

**Conclusion**

The development of the SysTec Smart Thermostat Prototype was a difficult yet awesome experience. It gave a practical application of embedded systems concepts and reinforced importance of more careful planning, modular design, and responsiveness in embedded software. The decisions made during this project have set a solid foundation for the next phase, which involves connecting the thermostat to the cloud and expanding its capabilities.