David Bortz

CS-350

Milestone 3

10/05/2025

The loop that processes the LED blinking needs to run in a separate thread to ensure non-blocking behavior, allowing the main program to continue executing other tasks without being hindered by the LED control logic. This is particularly important in applications where responsiveness is critical, as it enables simultaneous operations; thus, the program can handle user inputs, manage state transitions, and update displays while the LEDs blink as intended. Additionally, returning to the off state after each completed state action serves to reset the state machine to a known state, which simplifies state management and minimizes the risk of errors. By providing a clear endpoint for each action, the system can safely prepare for the next command without lingering effects from the previous state, avoiding unintended actions that could disrupt functionality.

To facilitate changing the messages available to the program, integrating serial communications can be beneficial. This involves setting up a serial interface, such as using `pySerial` in Python, to allow the program to receive commands or messages from an external source like a computer or microcontroller. A simple command protocol can be developed to enable specific messages or commands to be sent via the serial port, and a listener thread can be implemented to continuously read from the serial port, updating the active message when new data is received. This setup ensures that the system remains stable and responsive to user inputs.

Furthermore, utilizing a 16x2 display can provide valuable debugging information to users who don’t have access to the application console. By displaying important status messages, such as the current active message, state transitions, or error notifications, users can gain insights into the system's operation without needing console access. Implementing rotating messages or error indicators on the display allows for immediate feedback about the system’s status or any issues that arise. For example, displaying instructions or prompts directly on the screen can guide users on necessary actions, enhancing the overall usability of the application.