Jordan Mitchell

2/6/2025

Professor Moscon

Milestone 3 Lab Guide Questions

1. **Why does the loop that processes the LED blinking need to run in a separate thread?**

The LED blinking loop runs in its own thread so the program can continually respond to inputs while the LED sequence is active. If it wasn’t in a separate thread, the program would get stuck in the blinking process, and you wouldn’t be able to press the button to switch messages or stop the program until it finishes.

1. **What is the purpose of returning to the off state after each completed state action?**

Returning to the off state after each completed action (dot, dash, or pause) provides a consistent and smooth structure to the state machine. It serves as a "reset" point, making sure the LEDs are turned off and ready to transition to the next state. This design reduces ambiguity and makes sure the timing stays consistent, and the LEDs don’t accidentally overlap or get out of sync.

1. **How could you integrate serial communications to facilitate changing the messages available to the program?**

You can add serial communications which let the program take new user provided messages from a serial terminal. For example, the Raspberry Pi can listen for incoming messages sent via a serial terminal or another device. Once it's been received, the program can parse the input string, validate it, and update the current active message. This would make it easy to change messages without needing to edit the code or restart the program.

1. **How could you use the 16x2 display to provide debugging information to the user when they don’t have access to the application console?**

The 16x2 display is handy for showing debugging information in real time. You can display the current state (like dot, dash, or pause) as well as the message being sent. It can also show error messages or updates to the system, like "Button Pressed" or "Waiting for Input." This way, users without access to the console can still monitor the program's behavior and troubleshoot issues directly on the hardware.