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

The LED blinking loop needs to run in a separate thread to maintain its precise timing requirements while keeping the system responsive to the end user’s input. This allows the threading separation to ensure the state machine transitions between LED patterns occur with nearly perfect timing to button triggered message changes. My machine on the DisplayTest was always a little delayed from pressing button to changing output on the display.

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

By allowing the off state after each action, the machine can make a clean reset point to ensure it can transition between the different elements used in the Morse code. This precision timing allows each dash, dot, & pause to prevent overlapping signals. I guess you can call this a rhythm for the machine by defining it’s start & stop points while remaining stable.

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

One method to add serial communications is to enable communication between the main computer & another device. Each system can check for incoming messages between transmissions, storing them in a list while continuing to send the current message without delay. These changes could run in the background while a separate script keeps the LED timing.

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 screen can show basic error messages like "LED Problem" or "Button Stuck" when something goes wrong. Even if the display isn't working perfectly, these simple messages are easier to see than console text. For timing checks, the display could flash numbers like 500 to show milliseconds when testing dot durations. A quick button press could switch between normal mode and debug mode. Keep messages very short since the display is small. If the display completely fails, you can use the LED blink patterns for debugging instead - like 3 red flashes for an error. This backup method doesn't need the screen at all.