**Module 5 Lab Questions**

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

The loop that controls the LED blinking needs to run in a separate thread so that the rest of the program can still respond to other things while the LED is blinking. For example, if you press a button, the program needs to be able to notice that right away. If the LED blinking was running in the main program by itself, everything else would have to wait until it finishes. By putting the blinking loop in its own thread, it can run in the background while the program keeps checking for other actions, like button presses or screen updates.

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

Returning to the off state after each action helps keep the system organized and ready for the next instruction. It’s like resetting the machine so it knows what to do next. If we didn’t return to the off state, the system might get stuck in one part of the program and not work correctly the next time we want it to do something. Going back to the off state also makes the system safer and easier to understand, especially if there are problems or unexpected inputs.

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

Serial communication could be used to let another device, like a laptop, send new messages to the program. For example, a person could type a message on the computer and send it over a USB cable to the Raspberry Pi. The program could then read that message and update what it shows on the display or what message it sends out using Morse code. This would make it easier to update messages without needing to change the code every time.

**4. 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 can be used to show helpful messages or updates about what the program is doing. If something goes wrong or if the program reaches a certain point, the screen could display a short message explaining it. This is useful when you can’t see the console output, like when you're not connected to a monitor. For example, the screen could say "Button Pressed" or "Sending SOS" to let the user know what’s happening without needing to look at the code or computer logs.