**Lab Questions: Milestone Three**

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

The LED blinking loop runs in a separate thread so it doesn’t block the main program. This lets the Raspberry Pi still detect button presses while the message is being transmitted. If it wasn’t in a separate thread, the program would be stuck in the transmit loop and wouldn’t react to inputs until it was done.

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

Returning to the off state resets the system so it’s ready for the next part of the Morse message. It also makes sure the LEDs are turned off between dots, dashes, letters, and words, keeping the timing and blinking behavior clean and consistent.

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

You could use the serial port to receive a new message string from a connected computer or device. When a new message comes in through serial, you’d update the activeMessage variable in the state machine and display the new message on the LCD for confirmation. That way, you’re not limited to just “SOS” or “OK”; you could send any custom message over serial.

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 LCD could show messages like “Transmitting: SOS”, “Message Updated: OK”, or even short logs like “Button Pressed” or “LED: Dot”. This gives real time feedback without needing to open a terminal, which is useful if the Pi is used in a standalone or headless setup.