Reflection: RSLK - SysTick and Interrupts: Arnav Krishnan

1. I learned that SysTick interrupts allow the processor to handle tasks at a specific time without busy-waiting. We programmed the SysTick to fire every 1ms/480000 clock cycles at 48MHz, so it can manage the reflectance sensor read (Reflectance_Start() at 0 and Reflectance_End() at 1), independently of the main loop. dataValid lets the main loop know when there is new data.

2. One thing that was initially a little unclear but made a lot of sense after implementation is the splitting of the interrupt handler from the main loop. The interrupt handler handled the timing and sensor triggering, while the main loop does the interpretation and response, using dataValid to communicate information. Once the implementation made sense, the code felt very organized.

3. I'm guessing this would not be possible with our project, but what happens if the nature of your main necessitates processing data for over 1ms? dataValid would update before the main loop uses it. This seems like an edge-case scenario at worst, however, I am still curious as to how main and Systick would handle it.