Lab 5 Report
Topics and Questions

ECE341, University of Idaho

Fall 2019

Report

The following apply only to the report for Lab 5:

- In the Background section:
  - The majority (if not all) of the discussion in this section should be on interrupts:
    - How are interrupts invoked? Can you “call” an ISR like a typical C function? Why not? What happens if you try to?
    - Why is it necessary to include the special syntax for declaring an ISR?
    - If multiple interrupts are triggered at the same time, what determines which is processed first?
  - Explain when a section of code must be protected from interrupts. What is this section called? How do you implement this protection? What might be the consequences of such a section if it is entered after system initialization?
    - Hint: Think about a system that has multiple priorities, and a lower-priority ISR enters one of these protected sections. Will the higher-priority ISR execute at the right time? What is the term used for the situation when a lower-priority task “blocks” a higher-priority task?
- In the Testing and Verification section:
  - Use oscilloscope screen capture(s) to show the interrupt priority levels. Explain how the capture(s) indicate the correct nesting scheme. Hint: Trigger on the LED that is set during the CN ISR and change the button state; observe the behavior of the LED toggled in the Timer1 ISR.
  - Compared to the the previous lab (Lab 4), is the stepper motor running at a more accurate[1] speed? Why or why not?
- In the Conclusion:
  - Discuss the advantages and disadvantages of a real-time control system that uses polling versus an interrupt-based system. When might you choose one over the other?
  - What is the worst-case latency for a system that uses polling? What about an interrupt-based system?

[1] Accurate here means a more consistent period, where the variation between any two periods is smaller than it might have been in the last lab.