Jose Perales
Dr. Amy Larson
April 1, 2020

# Lab 8 Question

1. **What is the WCET for the following tasks? State whether you used the average time analysis described above or an oscilloscope/logic analyzer.**
   - Scheduler with 5 tasks.
   - Encoder code (do NOT run this in the ISR(PCINT) to determine WCET)
   - Potentiometer Readings

The scheduler with five tasks is 17.3, the encoder code is 6, and the potentiometer is 111.19

2. **What is the period of the encoder interrupt when the motor is running at 25% duty cycle and at a 90% duty cycle? Given your WCET above and these periods, what is the utilization of this task for each of the duty cycles?**

3. **If you run the scheduler every 1ms, what is the utilization of the scheduler? What about every 10ms? State whether or not you think 10ms is a viable period of the scheduler. Briefly justify.**

If one runs the scheduler every 1ms the utilization of the scheduler is 1.93%. If we run it every 10ms the utilization is .193%. 10ms is not a viable period of the scheduler because having the period being shorter is more efficient of the task.

4. **The potentiometer readings will be polled. What do you think is a good period for this task and why? Given that period, what is the utilization of polling the potentiometers?**

**Communication is an important part of the system -- both for the user to send commands to the system and for the system to report status. It is not included as part of this analysis partly because serial (using the dongle) is**

**interrupt-driven, therefore timing it is difficult. We will work on analyzing this task later.**

A good period for this task is 1ms. Utilization for this would be 11.119% and the excution is 11119 / 100000 = .11119ms

5. **Consider the various scheduling methods used throughout the labs and discussed in class. For each method below, discuss the control over the timing of that task, responsiveness of the system, and the impact on other tasks with respect to timing.**

- **Time-driven execution inside an ISR (e.g. toggle yellow every 250ms),**
  - A time-driving execution inside an ISR is using a system timer. Since an ISR is high priority it will run and trigger the "yellow led" right away. Other task with lower priorities will be interrupted.
- **Time-driven release managed by the scheduler,**
  - When the release time is reached, the scheduler will update the flag.
- **Event-driven using an external interrupt with execution inside an ISR (e.g. toggle green in response to a button press),**
  - As the system is reacting to the event immediately — it's postponing other task
- **Event-driven using an external interrupt to release a task (e.g. setting a flag in fresponse to a button press that signals a task in main() to execute).**
  - This is the same as the one above, the only has a higher priority.
- **Periodic polling of an event (e.g. Polling for a button press).**
  - As periodic polling requires its own execution time and it will require extra time. It will then get checked in main() also taking time. As it needs more time and steps than other, it affects the times of other tasks.