Edward Castillo
SENG 5831

Amy Larson, Ph.D.

Experiments and Report (to hand in)

Run a series of experiments as described below and answer the following questions. For each experiment,

- Zero all toggle counters (>za).
- Toggle the LEDs for approximately 30 seconds.
- Record the number of toggles for all LEDs (>pa).

1. Use your original version of toggling the red LED that uses forloops. Toggle all 3 at **1Hz**. (Do not type in any menu options while you are toggling until the **1 minute** is up). How good was your WCET analysis of the for loop? If it is very far off, adjust it. **Why did I not want you to use the menu while running the experiment?**

Based on my observations this worked great. The flash rate for the 3 lights were really close. I did not want to use the menu during this test because the menu will delay the loop while it is processing the input.

2. Use your **software timer to toggle the red LED**. Toggle all 3 at **1Hz**. Simply observe the final toggle count. All should be about 60 (maybe the red is off by 1). If this is not the case, you probably set something up wrong, and you should fix it.

This was working really well as well. My observations were about how you described in the question.

3. Set all LEDs to toggle at **2Hz (500ms)**. Place a 90ms busywait forloop into the ISR for the green LED. Toggle for 1 minute and record results. Now move the 90ms busywait forloop from the green into the ISR for the yellow LED. Toggle for 1 minute and record results. What did you observe? Did the busywait disrupt any of the LEDs? **Explain your results**.

The busywait in the green interrupt seemed to disrupt the RED led the most. The yellow and green seemed to be pretty close to unaffected. **R:98 G:119 Y:120**

Moving the busywait in the yellow interrupt disrupted the RED even more. **R:15 G:123 Y:120**

It seems that the yellow ISR has a greater impact on the main loop. Though checking the priorities, Green is the highest so I would have expected it's busy wait to have a greater impact.

4. Repeat #3, except use a 110ms busywait. You probably won't be able to use the menu functions. If not, report that, and discuss what you observed from the blinking. **Explain your results**.

In green it has no observable impact on the menu and the disruption was about the same as before. However, in yellow it disabled the menu

5. Repeat #3, except use a 510ms busywait. **Explain your results**.

At 510ms both locations disable the menu. In they both seem like it takes the 510ms to process each instruction. So to traverse the menu takes a couple of 510ms passes. However, I did not wait to see if that is true. It is amazing how much of a difference that makes.

---

6. Repeat #5 (i.e. 2Hz toggle with 510ms busywait), except place an sei() at the top of the ISR with the forloop in it. **Explain your results**.
So this round, I was able to set the leds to 2Hz in the green interrupt. The yellow locked out the menu completely.

So based on the exercises described it seems that the timing and coordination of the interrupts are what is being impacted. With no busywait, the system is able to keep up with the deadlines and remain pretty accurate on the counts. However, once the busywait is introduced it begins to impact the deadlines. This impact is dependant on the firing cycle time of the interrupt that has the busywait. If it is a higher priority interrupt, it will have less of an impact than if it is a lower priority interrupt. This is because the lower priority interrupt has to yield to the higher priority interrupt.

The other factor is when the time the busywait takes exceeds the firing resolution of the interrupt, it basically will queue up the interrupt before the current one has completed. The higher the priority will be the one that executes only. While a lower priority will yield to the higher.

Now the part that I wasn't able to confirm due to the menus being locked out, I would imagine that the the lower interrupt will finish, but the higher one will be queued up. Whereas the higher interrupt will perform the same.

In all above scenarios where the firing resolution was exceeded, I can only guess that the red is shut out completely. Basically because of the location of the interrupt.