**CSCI5143**

**Tianshu Lin**

**4724118**

**Lab1 Write-Up**

1. To calculate the worst case execution time, I used a toggle of LED to indicate the finished time of Hough transform. Setting the LED toggle at 5 seconds and found how many times the Hough transform had been run. Then divided that number by 100 because of 5s/(50ms)=100. Then I selected the number after the division as the times that Hough transform should run to guarantee the 50ms worst case execution time. In my program, the number was approximately 23.

The factors that affects the accuracy of WCET analysis is the accuracy of the time when counting the 5s time by observation LED.

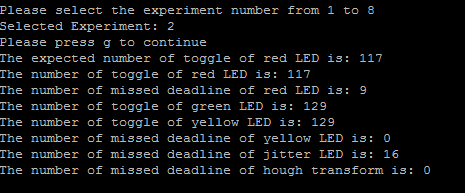
2. The periods of Hough Transform, red LED and yellow LED are all 100ms because I set the event of Hough transform flag and LEDs’ toggle event at 100ms. Jitter LED doesn't have the period because it is released by a random event. However, when delays are added into ISR, tasks are conflicted with each other, the real periods of tasks may have some little difference.

3.

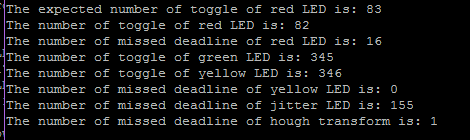
a: NO. There was no “drift” in the blinking of LEDs because I set all timers with accurate toggling frequency that made them toggle at exactly 1Hz. The drift could be caused by the setup of timer with inaccurate frequency thus affected the accuracy of the toggling rate.

b: NO. The LEDs were not on or off at the exact same time although their toggling frequencies are same. This was caused by the factor of when the setup of PWM and Timers are finished and when we firstly chose to toggle the LED in the main loop or ISR.

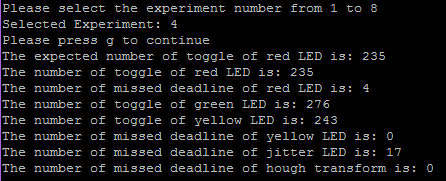
4.



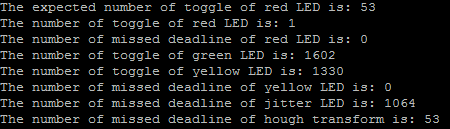
For experiment #2, since there is a 20ms delay in green LED ISR (entering in for each 200ms), the red LED toggling rate became smaller compared with green and yellow LED, and there were missed deadlines for red LED. However, there was no effect for yellow LED since the yellow LED’s ISR was entering in for each 25ms which larger than the 20ms delay.



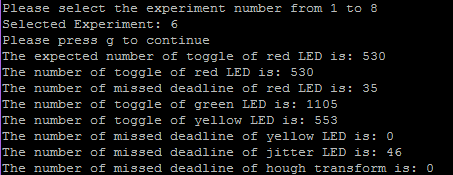
For experiment #3, since there was a 20ms delay in yellow LED ISR (entering for each 25ms), the red LED toggling rate became slower because it conflicts with the toggling time of red led. Also, there were missed deadlines for red LED.However, yellow LED did not get affected because its period for entering its ISR was 25ms which was longer than 20ms.



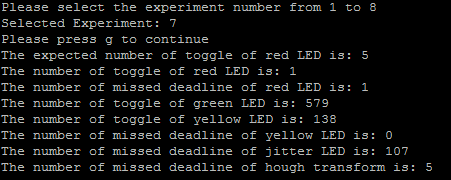
For experiment #4, since there was a 30ms delay in green LED ISR(entering in for each 200ms), which was longer than the period for entering yellow LED ISR, both red and yellow LED toggling rate became slower because it conflicts with the toggling time, and there were missed deadline for red LED.



For experiment #5, since there was a 30ms delay in yellow LED ISR(entering in for each 25ms), which was even longer than the period of yellow LED ISR itself, it would keep entering the yellow LED ISR and it would not go back to the main loop. Therefore, Red LED could not toggle (or just once at the beginning) because it never went back to main loop. Thus the missed deadline of red LED would be 0. Yellow LED would toggle at a slower rate.



For experiment #6, since there was a 105ms delay in green LED ISR(entering in for each 200ms), which was longer than the period for entering yellow LED ISR, both red and yellow LED toggling rate became slower because it conflicts with the toggling time, and there were missed deadline for red LED.



For experiment #7, since there was a 105ms delay in yellow LED ISR(entering in for each 25ms), which was even longer than the period of yellow LED ISR itself, it would keep entering the yellow LED ISR and it would not go back to the main loop. Therefore, Red LED could not toggle (or just once at the beginning) because it never went back to main loop. Thus the missed deadline of red LED would be 0. Yellow LED would toggle at a slower rate.

5. For experiment #8, the system froze after for a while because it kept entering the interrupt which caused the overflow of the stack.

6.

Analysis of the control over the timing of that task, responsiveness of the system, and the impact on other tasks with respect to timing.

1. Time-driven execution inside an ISR

Time-Keeper Task was implemented using Timer0, its count increased for each 1ms and it toggled red LED task when the count reached 100.

1. Time-driven release from within an ISR (e.g. Hough Transform Task),

Hough Transform Task was released by a flag for each 100ms inside Timer0, it did not affect other tasks.

1. External interrupt with execution inside an ISR (e.g. Communication Task, if inside ISR)

The PIN interrupt occurred when button A changed its condition. I set a variable which value is 2 so it meant after the button experienced both ‘press’ and ‘release’, the menu option would display.

1. Periodic polling for an event from within an ISR, then release of task (e.g. JITTER LED task).

Jitter LED Task was released by a flag for an random event inside Timer3(40Hz). It might affect the toggling of red LED because jitter LED had a delay and they both toggled inside main loop.