EE-443 LAB 7 Questions

Burak YÖRÜK

Question 1: `PrintByte(LCDtext,"",Atten);`

```
EnQueueLCDbuffer(0xC0); // LCDMoveCursor operation

pText = LCDtext;

while(*pText != 0x00){

    EnQueueLCDbuffer(*pText); // Send data to LCDbuffer

    pText++;

}
```

If Timer-1 ISR is called between these codes, the LCD may display wrong data. So we need to stop condition for ISR between them with using cli(); function and enable interrupt after the print Byte processes.

Question 2: Timer-2 COMPA defined as _VECTOR(7) in interrupt.h library Timer-1 CompA is defined as _VECTOR(11). Priority levels are arranged as number low to high. So, when Tİmer-2 comes to the process between Timer-1 interrupt it's also stops the interrupt and continue its higher priority interrupt. Interrupt-1 continues after the Timer-2 interrupt.

Question 3: If we look at the function Timer-2 interrupt which is used for printing data on LCD. And the enqueue operation puts data on LCD buffer. As mentioned Timer-2 has higher priority so when it comes it breaks other lower priority ISR. So, if it comes during enqueue operation, the buffer won't be loaded completely. Any lack of Buffer cause to be printing wrong data on the LCD screen. Hence, we can solve this problem with using Semaphores.

Question 4: The maximum response time is nearly 150us.

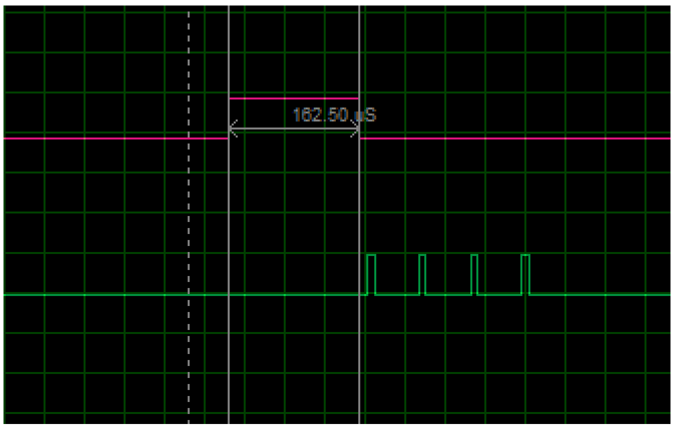

Figure 1: Maximum response time for Timer-1

Question 5 : In this program enqueue function takes the data to queue buffer, dequeue function writes these data to LCD. If there is no data in buffer, it is meaningless to call dequeue function again and again. It reduce the maximum response time for timer-1 and timer-2 interrupts. We can declare a global variable to say the function “there is data in queue buffer, let dequeue function display LCD”. The problem may be caused to arrange the priorities for both interrupts and main function. Interrupts may interrupt to each other or in main function interrupts can be called in wrong times. In order to prevent this we can set proper semaphores.