ECE-3226/CSCI -3451 EXPERIMENT #6
INTERRUPTS

OBJECTIVES:

1) To get familiar with setting up the interrupt routine addresses in the Interrupt Vector Table.
2) To get familiar with the enabling process of the interrupts.
3) To experiment interrupt service.
4) To experiment the priority order of the external interrupts (INT0, INT1, and INT2)

NOTE: Please turn in the following parts as pre lab.
Part1A, Part1B, Part1C, Part1D

PROCEDURE:

Part 1)

A) Write a main program such that:

i) Enables the INT0, INT1, and INT2 interrupts as negative edge triggered. (High to low transition). Leave the IVSEL bit at 0 (default) in the GICR register. This will allocate the Interrupt Vector Table at the beginning of the program memory.

ii) Counts 0 to 255 (back to 0) and outputs the binary equivalent to PORTA in intervals of one seconds continuously.

B) Write an interrupt routine for the INT0 interrupt that outputs 0x03 to PORTA in intervals of one seconds, five times. (Flashing 0x03 5 times)

C) Write an interrupt routine for the INT1 interrupt that outputs 0xC0 to PORTA in intervals of one seconds, five times. (Flashing 0xC0 5 times)

D) Write an interrupt routine for the INT2 interrupt that outputs 0xAA to PORTA in intervals of one seconds, five times. (Flashing 0xAA 5 times)

NOTE: Write a subroutine for implementing a one second delay. Use Simple delay loops using instruction execution times (1 to 2 μs).

NOTE: The initial body of the program is given below, with the Interrupt Vector Table already initialized to point to the interrupt routines.

```
;************************************************
; written by:
; date:
; file saved as:
; for AVR:                    ATmega32
; clock frequency:
;************************************************
; Program Function:   <describe purpose of program here>

; .device          ATmega32; don't need because it's in .inc file below
.nolist
.include      "C:\Program Files\Atmel\AVR Tools\AvrAssembler2\Appnotes\m32def.inc"
.list
;************************************************************
; setting up the interrupt routine addresses in the Interrupt Vector
; Table.
;************************************************************
     jmp       Init
     jmp        INT0_serv
     jmp        INT1_serv
     jmp        INT2_serv

;*********************
; main program
;*********************
.Org 0x30
Init:  ldi    r16 ,  low(ramend)          ;initialize the stack pointer
       out    spl ,   r16
       ldi    r16 ,  high(ramend)
       out   sph ,   r16

;   write main code here



;*****************************
;  Interrupt routine for INT0
;*****************************
INT0_serv:

;  code

RETI
```

```
;****************************
;  Interrupt routine for INT1
;****************************
INT1_serv:

;  code

RETI

;****************************
;  Interrupt routine for INT2
;****************************
INT2_serv:

;  code

RETI
```

Part2) Test your program in the lab in the following order and observe the results.
Connect INT0 (PD2) to SW0
Connect INT1 (PD3) to SW1
Connect INT2 (PB2) to SW2

1. Observe the counting. (Keep the interrupts deactivated.)

2. Apply a high to low transition to INT0 (push the SW0 and let go). State your observations in detail.
(wait until it returns to counting.)

3. Apply a high to low transition to INT1 (push the SW1 and let go). State your observations in detail.
(wait until it returns to counting.)

4. Apply a high to low transition to INT2 (push the SW2 and let go). State your observations detail.
(wait until it returns to counting.)

5. Apply a high to low transition to INT2. While INT2 is being serviced apply another high to low transition to INT2. State your observations in detail. (wait until it returns to counting.)

6. Apply a high to low transition to INT1. While INT1 is being serviced apply a high to low transition to INT0. State your observations in detail.

State the order the interrupts were serviced, and explain the reason for that order. Is INT0 able to interrupt INT1? Why or why not?
(wait until it returns to counting.)

7. Apply a high to low transition to INT0. While INT0 is being serviced apply a high to low transition to INT2. State your observations in detail. State the order the interrupts were serviced, and explain the reason for that order. Is INT2 able to interrupt INT0? Why or why not?
(wait until it returns to counting.)

8. Apply a high to low transition to INT2. While INT2 is being serviced first, apply a high to low transition to INT1 and next, apply a high to low transition to INT0. State your observations in detail. State the order the interrupts were serviced, and explain the reason for that order.
(wait until it returns to counting.)

9. Apply a high to low transition to INT0 and INT1 at the same time. The hardware connection will be explained in the lab. State your observations in detail. State the order the interrupts were serviced, and explain the reason for that order.
(wait until it returns to counting.)

Part3) Write a detailed conclusion from your observations of part 2.