Department of Computer Engineering

BLG 351E
Microcomputer Laboratory
Experiment Report

Experiment No : 5

Experiment Date : 25/11/2016

Group Number : Friday - 6

Group Members :

ID	Name	Surname
150160925	David	Musinguzi
150140051	Mertcan	Yasakçı
150160154	Gülnur	Kaya
150130035	Berkan	Öztekin

Laboratory Assistant : Neziha Akalın

1 INTRODUCTION

In this experiments we enhance our understanding of the 7-segment displays and initializing interrupt.

In the first part of the experiment we show decimal integers 0-9 on the 7- segment display and the second part we implement an interrupt subroutine that enables the program to switch between counting upwards and downwards.

2 EXPERIMENT

2.1 PART 1 COUNTER PROGRAM

The first part we had to figure out which LEDs to switch on the show the number 0-9; we implanted the following: 00111111b, 00000110b, 01011011b, 01001111b, 01100110b, 01101101b, 01111101b, 00000111b, 01111111b, 01101111b

Then we put the binary representations into an array called “element”, after we wrote a program that counts from 0-9.

```
Setup       mov.b  #0,&P1OUT
            mov.b    #FFh,&P1DIR
            mov.w  #element,R6
            mov.w  #element,R7
            mov.b    #10,R8

Loop1       mov.b  @R6,&P1OUT
            inc.w    R6

            call     #Delay

            dec.b  R8
            jnz    Loop1
            mov.w    R7,R6
            mov.b  #10,R8
            jmp      Loop1

Loop2       mov.w  #element,R6
            mov.b    #9,R8
            mov.b  @R6,&P1OUT
            dec.w    R6
            call   #Delay

            inc.b  R8
            cmp.b  #9,R8
            jz           Loop1
            jmp          Loop2

Delay       mov.w    #0Ah ,R14
L2          mov.w    #07A00h ,R15
L1          dec.w    R15
            jnz      L1
            dec.w    R14
            jnz      L2
```

```
            ret




;Integer array

element             .byte  00111111b, 00000110b, 01011011b, 01001111b, 01100110b,
01101101b, 01111101b, 00000111b, 01111111b, 01101111b
lastElement
```

2.2 PART 2 INTERRUPT SUBROUTINE

In this section we were required to implement an interrupt subroutine, to enhance our main program by enabling it to count both upwards and downwards.

We defined a Boolean variable in our program that which represents the direction of the counting.

So, then in our main loop, we checked the value of the Boolean variable and the program decide whether to count upwards or downwards.

We simply toggle the value of the variable in our interrupt faction between 1 and 0 by XOR.

```
init_INT    bis.b #040h,&P2IE    ; enable  interrupt  at P2.6
            and.b #0BFh ,& P2SEL       ; set 0 P2SEL.6
            and.b #0BFh ,& P2SEL2      ; set 0 P2SEL2 .6

             bis.b    #040h,& P2IES     ; high -to -low  interrupt  mode
             clr    &P2IFG          ; clear  the  flag
             eint            ; enable  interrupts




;code---
Setup        mov.b  #0,&P1OUT
             mov.b    #255,&P1DIR
             mov.w  #element,R6
             mov.w  #element,R7
             mov.b    #10,R8
             mov.b  #0h, R10

main         cmp.b  #0h, R10
             jnz          Loop2


Loop1        mov.b  @R6,&P1OUT
             inc.w    R6

             call     #Delay

             cmp          #lastElement, R6
             jnz    main
             mov.w  #element,R6
             jmp      main
```

```
Loop2       mov.b  @R6,&P1OUT
            dec.w    R6
            call   #Delay
            mov.w  #element,R8
            dec           R8
            cmp           R8,R6
            jnz           main
            mov.w  #lastElement,R8
            dec           R8
            mov.w  R8,R6
            jmp           main

Delay       mov.w    #0Ah ,R14
L2          mov.w    #07A00h ,R15
L1          dec.w    R15
            jnz      L1
            dec.w    R14
            jnz      L2
            ret

ISR         dint        ; disable  interrupts

;code--
            xor.b  #1h, R10
            clr    &P2IFG          ; clear  the  flag

            eint        ; enable  interrupts
            reti        ; return  from  ISR

;Integer array

element            .byte  00111111b, 00000110b, 01011011b, 01001111b, 01100110b,
01101101b, 01111101b, 00000111b, 01111111b, 01101111b
lastElement

;-------------------------------------------------------------------------------
; Stack Pointer definition
;-------------------------------------------------------------------------------
            .global __STACK_END
            .sect   .stack

;-------------------------------------------------------------------------------
; Interrupt Vectors
;-------------------------------------------------------------------------------
            .sect   ".reset"                ; MSP430 RESET Vector
            .short  RESET
            .sect   ".int03"    ; Port  Interrupt  Vector
            .short   ISR
```

3 CONCLUSION

First we figured out how to light up the 7-segment display, then wrote a program to manipulate it. The hard part came when we had to implement an interrupt and figure out a way to count backwards; we figured it out by using the array point at the beginning and end to count upwards and downwards.

In conclusion in this experiment, we learnt how to manipulate the 7-segment display and how to implement an interrupt function and always widened out understand and use for arrays and memory pointers further.