Department of Computer Engineering

BLG 351E
Microcomputer Laboratory
Experiment Report

Experiment No : 6

Experiment Date : 10.12.2015

Group Number : 2

Group Members :

ID	Name	Surname
040100055	Emir	Bilgin
150120251	Cem Yusuf	Aydoğdu
150120121	Hüseyin	Tosun
Click here to enter text.	Click here to enter text.	Click here to enter text.

Laboratory Assistant : Bilge Süheyla Akkoca

1 INTRODUCTION

This experiments aimed to enhance the practical experience about driving 7-segment displays and initializing interrupts.

2 REQUIREMENTS

2.1 PART1

In the first part of the experiment, we wrote a main program which counts from 0 to 9 in ten seconds and shows the value of the counter at one of the digits of the 7-segment display. Firstly we initialized GPIO Port 1 and Port 2 to activate and used 7-segment display. Then, we wrote a loop which increments a counter and waits one-second at each iteration.

```
                    bis.b   #11111111b,&P1DIR   ;set all button in P1 port
                    bis.b   #00001000b,&P2DIR
                    bis.b   #00001000b,&P2OUT   ;set P2.3

                    mov           #array,R5     ;load array to R5
                    mov           #00h, R6      ;control register for counting

Loop                mov.b         @R5,&P1OUT    ;lighting output on 7-segment display
                    call          #Delay        ;1 second delay
                    inc           R5            ;next element in array
                    inc.b         R6            ;increment control register
                    cmp           #0Ah,R6       ;comparing control register for break loop
                    jne           Loop          ;counting 0 to 9
                    mov           #00h, R6      ;clearing control register for start again
                    mov           #array,R5     ;load array to R5
                    jmp           Loop          ; return back to 0 and count to 9 again

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

```
                        jnz L2
                        ret


array           .byte 00111111b, 00000110b, 01011011b, 01001111b, 01100110b, 01101101b,
01111101b, 00000111b, 01111111b, 01101111b ;contains 10 values

lastElement             ;array elements for displaying on 7-segment display
```

2.2 PART2

In this part, we write an interrupt subroutine and enhance your main program so that our program able to count upwards OR downwards by an external interrupt. First, in our main loop, we checked the value of this variable and decide to increment or decrement the value of the counter. After that, we activated a mask-able interrupt in one of the GPIO Port Pins. Then, we stored the address of our interrupt subroutine to the interrupt vector of the GPIO Port 2.

```
                        bis.b   #11111111b,&P1DIR       ;set all button in P1 port
                        bis.b   #00001000b,&P2DIR
                        bis.b   #00001000b,&P2OUT       ;set P2.3


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


                        mov     #array,R5       ;load array on R5
                        mov     #00h, R6        ;control register for counting
                        clr     R7              ;control register for increment or decrement operation


Loop                    mov.b   @R5,&P1OUT          ;lighting output on 7-segment display
                        call    #Delay              ;1 second delay
                        cmp     #0h, R7             ;comparing for decrement of increment op.
                        jne     Dec


Inc                     inc     R5              ;next element in array
                        inc.b   R6              ;incrementing counter
```

```
                cmp     #0Ah,R6         ;comparing counter
                jne     Loop            ;if not equal 0 then continue loop
                mov     #00h, R6        ;setting counter 0 for start to loop again
                mov     #array,R5       ;loading array on R5
                jmp     Loop            ; when loop finished start again

Dec             dec     R5              ;preveious element in array
                dec.b   R6              ;decresing counter
                cmp     #00h,R6         ;comparing control register
                jne     Loop            ;doing loop until control register is zero
                mov     #array,R5       ; write zero
                mov.b   @R5,&P1OUT ; lighting output on 7-segment display
                call    #Delay          1 second delay
                mov     #09h, R6        ;setting counter 9 for decresing
                mov     #array,R5       ;loading array to R5
                add     #09h, R5        ;starting with last element of array
                jmp     Loop            ;continue loop

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
                clr &P2IFG      ;clear the flag
                xor #01h, R7    ;enabling interrupt choosing decrement or increment operation
                eint            ; enable interrupts
                reti            ; return from ISR
```

```
array		.byte	00111111b, 00000110b, 01011011b, 01001111b, 01100110b, 01101101b, 01111101b, 00000111b, 01111111b, 01101111b ;contains 10 values

lastElement
```

3 CONCLUSION

In this experiment we learned how to enhance the practical experience about driving 7-segment displays and initializing interrupts.