

İTÜ



Department of Computer Engineering

BLG 351E Microcomputer Laboratory Experiment Report

Experiment No : 7
Experiment Date : 09.12.2016

Group Number : Friday - 3
Group Members :

ID	Name	Surname
150130109	Güllü	Katık
150140142	Gamze	Akyol
150130136	Mehmet Barış	Yaman

Laboratory Assistant : Emrullah Gazioğlu

1. INTRODUCTION

In this experiment we tried to write

“ITU - Comp. Eng.

MC Lab. 2016”

to the MSP 430 LCD Display by writing 4 different subroutines.

2. EXPERIMENT

MSP430 microcontroller is connected only 4-bits of lcd, while LCD has 8-bits and 4-bits mode. Because of that, we initialized the LCD displayer according to flowchart given in the booklet. We divided the 8-bit data to 4-bit data and sent this to LCD displayer. Every time when the data was sent to LCD, triggerEN subroutine was called.

The code of this experiment given below:

SetupP1	mov.w	#string, R5	;R5 pointer
	mov.w	#11111111b, &P1DIR	; Initializing P1 as output
SetupP2	mov.w	#11111111b, &P2DIR	; Initializing P2 as output
	clr	&P2SEL	
	clr	&P2SEL2	
InitStart	call	#Delay	
	mov.w	#00110000b, &P1OUT	
	call	#triggerEN	
	call	#Delay	
	mov.w	#00110000b, &P1OUT	
	call	#triggerEN	
	call	#Delay	
	mov.w	#00110000b, &P1OUT	
	call	#triggerEN	

BLG 351E Microcomputer Laboratory – Experiment Report

```
call    #Delay
mov.w   #00100000b, &P1OUT
call    #triggerEN
```

```
call    #Delay
mov.w   #00100000b, &P1OUT
call    #triggerEN
mov.w   #11000000b, &P1OUT
call    #triggerEN
```

```
call    #Delay
mov.w   #00000000b, &P1OUT
call    #triggerEN
mov.w   #10000000b, &P1OUT
call    #triggerEN
```

```
call    #Delay
mov.w   #00000000b, &P1OUT
call    #triggerEN
mov.w   #00010000b, &P1OUT
call    #triggerEN
```

```
call    #Delay
mov.w   #00000000b, &P1OUT
call    #triggerEN
mov.w   #01110000b, &P1OUT
call    #triggerEN
```

```
InitEnd    call    #Delay

mov.w   #00000000b, &P1OUT
call    #triggerEN
```

BLG 351E Microcomputer Laboratory – Experiment Report

	mov.w	#11110000b, &P1OUT
	call	#triggerEN
End	call	#Delay
Set	mov.w	#10000000b, &P1OUT
	call	#triggerEN
	mov.w	#00000000b, &P1OUT
	call	#triggerEN
Write	call	#Delay
	mov.w	#10000000b, &P2OUT
	mov.b	@R5,R6
	mov.b	@R5,R7
	and.b	#11110000b,R7
	and.b	#00001111b,R6
	rlc.b	R6
	rlc.b	R6
	rlc.b	R6
	rlc.b	R6
	mov.b	R7, &P1OUT
	call	#triggerEN
	mov.b	R6, &P1OUT
	call	#triggerEN
	inc	R5
	cmp.b	#00h, R5
	jeq	finish
	jmp	Write
	call	#Delay
finish	nop	
Delay	mov.w	#0Ah,R14

```

L2          mov.w      #07C00h,R15  ; WAIT SUBROUTINE FOR DELAYS
L1          dec.w      R15
           jnz         L1
           dec.w      R14
           jnz         L2
           ret

triggerEN   bis.b      #01000000b, &P2OUT    ;set as 1 the 2nd bit of P2OUT
           bic.b      #01000000b, &P2OUT    ;clear the 2nd bit of P2OUT
           ret

           .data

string      .byte      "ITU - Comp. Eng.", 0Dh, "MC Lab. 2016", 00h
    
```

3. CONCLUSION

In this experiment, the hardest part was writing the initializing part for the LCD Display. Additionally, the difference between sending commands and sending initialized instructions came us hard to understand. That made us to think that we might have mistake in initializing LCD Display.

After the experiment, we have thought that our mistake was in the command:

```
cmp.b #00h, R5
```

Since, R5 was a pointer (initialized in the first command of the program) which holds the indexes in the array ‘string’ , we have to access the elements of R5 by writing ‘@’ in the beginning of the command according to the MSP 430 Board Assembly syntax. Therefore, the true command has to be:

```
cmp.b #00h, @R5
```

In the experiment, we saw that the program wrote

“ITU - Comp. Eng.

MC Lab. 2016”

to the screen as expected, but with various characters before and after the sentence. Since the comparison between R5 and #00h was wrong, the comparison result was wrong and did not work as expected. The, loop did not stop and R5 took invalid characters depending on the memory slot’s values.

Moreover, by the difficulties we have faced, we have learned how to write words to the MSP 430 Board Display and how to initialize the MSP 430 Board accordingly. We have learned how to trigger enable inputs, how the connection between P1, P2 and the board is obtained.