Department of Computer Engineering

BLG 351E
Microcomputer Laboratory
Experiment Report

Experiment No : 7

Experiment Date : 27/11/17

Group Number : Monday - 17

Group Members :

ID	Name	Surname
150150302	İrem Nur	Demirtaş
150160706	Merve Elif	Demirtaş
150140719	Cemal	Türkoğlu
150140722	İbrahim	Dolapçı

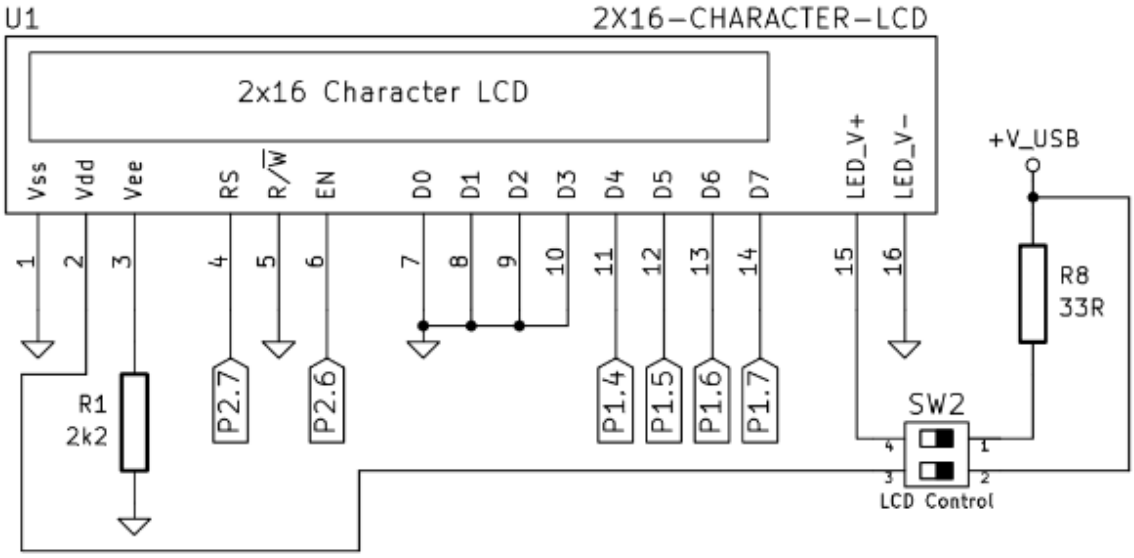
Laboratory Assistant : Ahmet Arş

1 Introduction

The aim of the experiment is implementing the program that drives 16x2 dot matrix LCD on the experiment board. The program is going to use a predefined char array as an input and display the string using LCD.

2 Experiment

Here is the connection between the LCD and the MSP430 LaunchPad is given.



There are two things that is required to do in this lab: Firstly configuring the LCD display in order to communicate in 4-bit mode. Secondly sending 8-bit ASCII characters as nibbles (4 bits) to display using the specific instruction. We are requested to implement a print function which uses char arrays as inputs and show them on the LCD display. This is our data and expected output.

```
1            .data
2 string     .byte "ITU - Comp. Eng.",0Dh,"MC Lab. 2017",00h
```

Here is the subroutines that we wrote in the program:

initLCD: The steps of initialization and configuration are implemented in this subroutine. Initially, the LCD works in 8-bit mode until you configure it otherwise. So, we should send your first commands accordingly.

```
1. initLCD      mov.b #00000000b,&P2OUT
2.              mov.b #00110000b,&P1OUT
3.
4.              ; her delayden once R13 e deger ata
5.              mov.w   #0c33h,R13
6.              call    #delay
7.              call    #triggerEN
8.
9.              mov.b #00110000b,&P1OUT
10.                 mov.w   #080h,R13
11.                 call    #delay
12.                 call    #triggerEN
13.
14.                 mov.b #00110000b,&P1OUT
15.                 mov.w   #04h,R13
16.                 call    #delay
17.                 call    #triggerEN
18.
19.                 mov.b #00100000b,&P1OUT
20.                 mov.w   #04h,R13
21.                 call    #delay
22.                 call    #triggerEN
23.
24.                 mov.b #00100000b,&P1OUT
25.                 mov.w   #04h,R13
26.                 call    #delay
27.                 call    #triggerEN
28.
29.                 mov.b #10000000b,&P1OUT
```

```
30.                     mov.w   #02h,R13
31.                     call    #delay
32.                     call    #triggerEN
33.
34.                     mov.b #00000000b,&P1OUT
35.                     mov.w   #02h,R13
36.                     call    #delay
37.                     call    #triggerEN
38.
39.                     mov.b #10000000b,&P1OUT
40.                     mov.w   #5eh,R13
41.                     call    #delay
42.                     call    #triggerEN
43.
44.                     mov.b #00000000b,&P1OUT
45.                     mov.w   #02h,R13
46.                     call    #delay
47.                     call    #triggerEN
48.
49.                     mov.b #00010000b,&P1OUT
50.                     mov.w   #02h,R13
51.                     call    #delay
52.                     call    #triggerEN
53.
54.                     mov.b #00000000b,&P1OUT
55.                     mov.w   #5eh,R13
56.                     call    #delay
57.                     call    #triggerEN
58.
59.                     mov.b #01100000b,&P1OUT ; left shift ,right shift
60.                     mov.w   #5eh,R13
61.                     call    #delay
```

```
62.                     call    #triggerEN
63.                     ret
```

sendCMD: Works in 4 bit mode, first loads the upper nibble of the command to the output port and calls trigerEN function to send the data to LCD, then repeats the same steps for the lower nibble.

```
1  sendCMD      mov.b #11110000,R6          ; R5 function parameter
2                                           ; takes command
3               and.b R5,R6
4               mov.b R6,&P1OUT
5               call  #triggerEN
6
7               rla.b R5
8               rla.b R5
9               rla.b R5
10              rla.b R5
11              mov.b R5,&P1OUT
12              call  #triggerEN
13              ret
```

sendDATA: Works in 4 bit mode, differs from sendCMD in terms of RS input, similarly it first it loads the upper nibble of the command to the output port and calls trigerEN function to send the data to LCD, then repeats the same steps for the lower nibble.

```
1               sendDATA bis.b  #10000000b,&P2OUT
2                        mov.b  0(R7),R8
3                        and.b  #11110000b,R8
4                        mov.b  R8,&P1OUT
5                        call   #triggerEN
6
7                        rla.b  R8
8                        rla.b  R8
9                        rla.b  R8
```

```
10                  rla.b   R8
11                  mov.b   R8,&P1OUT
12                  call    #triggerEN
13                  bic.b   #10000000b,&P2OUT
14                  ret
```

delay: busy-wait (loop) function to create necessary delays which is mandatory for LCD to function properly.

```
1.delay     mov.w   #0Ah,R14 ; R13 delay miktari , parametre
2.L2        mov.w   R13,R15
3.L1        dec.w   R15
4.          jnz     L1
5.          dec.w   R14
6.          jnz     L2
7.          ret
```

trigerEN: first it changes the value of EN to high (1) then it changes it back to low (0).

```
1.  triggerEN   bis.b   #01000000b,&P2OUT   ;toggle enable
2.              bic.b   #01000000b,&P2OUT
3.              ret
```

3 CONCLUSION

In this experiment, we struggled with configuring the LCD very long time. At the end the assistant who is checking our codes agreed that there seems no problem but we could not manage to print the result to led.