Department of Computer Engineering

BLG 351E
Microcomputer Laboratory Experiment Report

Experiment No : 8

Experiment Date : 08.12.2017

Group Number : Friday - 13

Group Members :

ID	Name	Surname
150160537	OKAN	YILDIRIM
150160546	HASAN EMRE	ARI

Laboratory Assistant : Yusuf Hüseyin Şahin

1 INTRODUCTION

In this experiment, we implement a program that displays the value on the LCD screen from the thermometer on DS18B20 by using MSP430 Education Board, MSP430G2553 microcontroller and its assembly language. Before the experiment, we studied on Background information and Experiment sheet in detail. We used the information that initialize to LCD from previous experiment. For using the thermometer, we read the datasheet of DS18B20 to ready experiment. In particular, we studied on thermometer initialization, ROM commands, function commands, 1 wire bus protocol and resolution of measure the thermometer.

2 EXPERIMENT

This experiment was more difficult and different other experiments. Timing and delays are vital for this experiment and it is required to read and understand the datasheet of DS18B20 well. Especially, understanding 1-Wire Communication Protocol is important to complete this experiment. We did our preliminary work, yet we cannot understand the some parts of this document actually and we try to implement as we understood.

Firstly, we implemented variety of delays such as 60 microseconds, 480 microseconds, 800 milliseconds, , 200 milliseconds, 100 milliseconds according to the datasheet of DS18B20. Then, we implement the LCD part of the program in similar way to the Experiment 7. Then we try to implement ***initTemp*** subroutine which is initialization procedure where you will implement the reset and presence pulse operations, ***Write*** subroutine which is used to write either ROM commands (or function commands and ***Read*** subroutine to read a bit from the 1-wire bus. We did something wrong about calling correct delays unfortunately. Therefore, Our program is did not work correctly. However, order of calling commands and functions in main part is correct absolutely. We try to store read data with stack and then display on the LCD.

Our program and detailed description is given below:

```
27  Setup           clr.b &P2SEL                ;clear the flags
28                  clr.b &P2SEL2
29                  mov.w #300h,SP              ; initialize the stack
30                  mov.b #0ffh, &P1DIR         ;set all P1DIR
31                  mov.b #11010000b, &P2DIR   ;set only uppest two P2DIR   mov.b #0ffh, &P2DIR
32
33  Main            call #initLCD               ;initialize LCD
34                  call #Open                  ;Open LCD
35                  call #initTemp              ;initialize and reset DS18B20
36
37                  mov.b #0CCh,R13             ;We only use single slave so that we send command to DS18B20 for skipping ROM
38                  call  #Write                ; Write this command
39
40                  mov.b #044h,R13             ; We send the command to start to measure and convert the temperature
41                  call  #Write
42
43                  call #initTemp              ; initialize and reset DS18B20  again
44
45                  mov.b #0CCh,R13             ;We only use single slave so that we send command to DS18B20 for skipping ROM
46                  call #Write
47
48                  mov.b #0BEh,R13             ; We send the command to read scratchpad
49                  call #Write
50
51                  call #Read                  ; Read the LSB of data from DS18B20
52
53                  push.w R13                  ; r13=0000LSB push to stack
54                  call #Read                  ; Read the MSB of data from DS18B20
55                  swpb R13                    ;swap byte for adding MSB part R12=MSB0000
56                  add.w @SP+,R13              ; R13=MSB+LSB
57
58                  call #Print
59                  jmp Main
60
```

```
;;;;;;;;;;;;;;;Print subrouitineis similar to the previous experiment7;;;;;;;;;;;;;;;
Print           call #sendDATA
                mov.b R13, R6           ;assign R13 (temperature data) to R6 and display it on the LED
                cmp.b #000h, R6         ;control
                jz end                  ;if finish jump to finish and end up print
                cmp.b #00Dh, R6         ;control
                jz nextLine             ; if char=\n go to nextLine label
                mov.b R6, &P1OUT        ; send the data to the LCD, yet only upper 4 bits
                call #triggerEn         ; enable it
                rla.b R6                ; for sending other 4 bits of data
                rla.b R6
                rla.b R6
                rla.b R6
                mov.b R6, &P1OUT        ;send remain 4 bits to the LCD
                call #triggerEn         ;enable it
                call #Delay100          ; delay needed
                inc.w R5
                jmp Print
nextLine        call #sendCMD           ; write on second line
                mov.b #10100000b,&P1OUT;set the DDRAM address to 40 - upper 4bits
                call #triggerEn
                mov.b #10000000b,&P1OUT;lower 4bits
                call #triggerEn
                call #Delay100
                inc.w R5                ; next character, remain characters are printed on second line
                jmp Print
end             mov.b #000h, &P1OUT
                ret
;;;;;;;;;;;;;;;;;;;;;initLCD is similar to the previous experiment7;;;;;;;;;
initLCD     mov.b   #00000000b,&P2OUT ;clear RS so that send commmand to LCD
            call #triggerEN
            call #delay100      ;more than 100ms

            mov.b   #00110000b,&P1OUT   ;Special case of 'Function Set' (lower four bits are irrelevant)
            call #triggerEN
            call #delay100      ; more than4.1 ms

            mov.b   #00110000b,&P1OUT ;Special case of 'Function Set' (lower four bits are irrelevant)
            call #triggerEN
            call #delay100  ;more than 100us

            mov.b   #00110000b,&P1OUT;Special case of 'Function Set' (lower four bits are irrelevant)
            call #triggerEN
            call #delay100  ;more than 100us

            mov.b   #00100000b,&P1OUT ;nitial 'Function Set' to change interface (lower four bits are irrelevant)
            call #triggerEN
            call #delay100  ;more than 100us

            ;*********************8 bit to 4 bit ***************************

            mov.b   #00100000b,&P1OUT ;upper 4bits
            call #triggerEN
            mov.b   #10000000b,&P1OUT   ;lower 4bits 'Function Set' (I = 1, N=1 it means I use secondline also )
            call #triggerEN
            call #delay100  ;;more than 100us

            ;upper 4bits
            mov.b   #00000000b,&P1OUT   ;'Display ON/OFF Control'   (D=1, C=0, B=0 )
            call #triggerEN
            mov.b   #10000000b,&P1OUT   ;lower 4bits
            call #triggerEN
            call #delay100

            ;upper 4bits
            mov.b   #00000000b,&P1OUT   ;'Clear Display' (no configurable bits )
            call #triggerEN
            mov.b   #00010000b,&P1OUT   ;lower 4bits
            call #triggerEN
            call #delay100

            mov.b   #00000000b,&P1OUT ;upper 4bits -- Entry mod set I/D =1 cursor move direction
            call #triggerEN           ;                           S=0 not shift the display
            mov.b   #01100000b,&P1OUT   ;lower 4bits
            call #triggerEN
            ret
```

```
;;;;;;;;;;;; Send CMD, sendDATA and triggerEn subrouitines are similar to the previous experiment7;;;;;;;
sendCMD         mov.b #000h, &P2OUT
                ret

sendDATA        mov.b #10000000b, &P2OUT        ; These are used in initLCD and Print when needed
                ret

triggerEn       bis.b #01000000b, &P2OUT
                bic.b #01000000b, &P2OUT
                ret
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;DS18B20 PART;;;;;;;;;;;;;;;;;;;;;;;
initTemp        mov.b #00010000b, &P2DIR
                mov.b #00000000b, &P2OUT ;reset pulse
                call  #delay480          ; delay 480 us
                mov.b #00000000b, &P2DIR ;release
                call  #delay60
isread          cmp.b #00000000b, &P2IN ; check for response
                jnz   isread
                ret

Write           mov.w #8h,R7 ;for each bit of data
A0              bis.b #00010000b,&P2DIR ;
                call #delay60 ; ~ 60us delay
                rrc.b R13 ;rotate
                jc A1 ;check for carry bit
                jmp A2 ;
A1              bis.b #00010000b,&P2DIR ;
                bic.b #00010000b,&P2DIR ;
                call #delay60 ; ~ 60us delay
                jnz A0 ;
                ret ;
                ;IN similar way to Write
Read            mov.w #8h,R7 ;for each bit of data
B0              bis.b #00010000b,&P2DIR ;
                bic.b #00010000b,&P2DIR ;
                bit.b #00010000b,&P2IN ; check for input==1
                rrc.b R13 ;rotate
                call #delay60 ;  60us delay
                dec.w R7 ;
                jnz B0 ;
                ret ;
```

```
186  ;;;;;;;;;;;;;;;;;;;;;;;;DELAYS;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
187  Delay100        mov.w #001h, R14                ;Delay 100ms
188  L1              dec.w R15
189                  jnz   L1
190                  dec.w R14
191                  jnz   L2
192                  ret
193  Delay200        mov.w #002h, R14                ;Delay 200ms
194  L20             mov.w #07A00h, R15
195  L21             dec.w R15
196                  jnz   L21
197                  dec.w R14
198                  jnz   L20
199                  ret
200
201  Delay800        mov.w #008h, R14                ;Delay 800ms
202  L10             mov.w #07A00h, R15
203  L9              dec.w R15
204                  jnz   L9
205                  dec.w R14
206                  jnz   L10
207                  ret
208
209  delay3          mov.w #00Ah, R14
210  L6              mov.w #07A00h, R15 ;1sn
211  L5              dec.w R15
212                  jnz L5
213                  dec.w R14
214                  jnz L6
215                  ret
216
217
218  delay60         mov.w #0001h, r14
219  L8              mov.w #0012h, r15 ;   60 micro saniye
220  L7              dec.w r15
221                  jnz  L7
222                  dec.w r14
223                  jnz  L8
224                  ret
225
226  delay480        mov.w #008h, r14
227  L4              mov.w #0012h, r15   ;  about 480 micro saniye
228  L3              dec.w r15
229                  jnz  L3
230                  dec.w r14
231                  jnz  L4
232                  ret
233  ;-------------------------------------------------------------------------------
```

3 CONCLUSION

We could not complete whole experiment. The timing was very important because of the 1 wire bus used in this experiment. We were configure the timing correctly. We had difficulty and make some mistake in this part. Perhaps more specific and descriptive information could be added to the background information from the datasheet of DS18B20. We needed more time for better understood of the documentations before the experiment and also for doing the experiment in class.