

İTÜ



Department of Computer Engineering

BLG 351E Microcomputer Laboratory Experiment Report

Experiment No : 6
Experiment Date : 24.11.2017

Group Number : Friday - 13

Group Members :

ID	Name	Surname
150160537	OKAN	YILDIRIM
150160546	HASAN EMRE	ARI
150150906	MAREVA	ZENELAJ*

*Mareva is member of another group normally, yet we did this experiment 6 together for once due to the absence of other members of the Marevas's group.

Laboratory Assistant : Yusuf Hüseyin Şahin

1 INTRODUCTION

In this experiment we design a chronometer with centisecond precision by using MSP430 Education Board, MSP430G2553 microcontroller and its assembly language. We enhanced the practical experience. Before the experiment, we studied on MSP430 User Guide – Timer A chapter document and Background information on experiment sheet. We get familiar with using 7-segment display and interrupts. We did preliminary work.

2 EXPERIMENT

2.1 PART 1

In this program of the experiment, we write code to lit determined digits of 7-segment display panel in an infinite loop. We lit 0123 digits on 7-segment display.

Our program and detailed description is given below:

Setup	<pre>bis.b #11111111b, &P1DIR bis.b #00001111b, &P2DIR</pre>
Mainloop	<pre>mov.w #00001000b, &P2OUT mov.w #01001111b,R5 mov.b R5,&P1OUT clr &P1OUT mov.w #00000100b, &P2OUT mov.w #01011011b,R5 mov.b R5,&P1OUT clr &P1OUT mov.w #00000010b, &P2OUT mov.w #00000110b,R5 mov.b R5,&P1OUT clr &P1OUT mov.w #00000001b, &P2OUT mov.w #00111111b,R5</pre>

```
mov.b R5,&P1OUT  clr &P1OUT

jmp Mainloop
```

Firstly, we initialized Port 1 and Port 2 to activate and use 7-segment display. Then, we assigned the values 0,1,2 and 3 to the determined 7-segment display panel simultaneously and turn on the digits on the LEDs. After the each lit we clear the P1OUT. Therefore, 0,1,2 and 3 are lit on the 7-segment panel simultaneously in an infinite loop. It happens too fast that all panels seem to be lit at the same time.

2.2 PART 2

In this part, we implemented a time interrupt subroutine, interrupt subroutine and convert subroutine by enhancing the main part of the previous part above in order to build the chronometer. We added required parts for using interrupt subroutine. We followed the steps given on the experiment sheet. Especially, MSP430 User Guide – Timer A chapter is very important to construct timer interrupt program.

Our code is given below:

```
;-----
; MSP430 Assembler Code Template for use with TI Code Composer Studio
;
;
;-----

.cdecls C,LIST,"msp430.h"    ; Include device header file

;-----

.def  RESET                ; Export program entry-point to
                        ; make it known to linker.

;-----

.text                    ; Assemble into program memory.

.retain                  ; Override ELF conditional linking
                        ; and retain current section.

.retainrefs              ; And retain any sections that have
                        ; references to current section.

.data

seconds      .byte 00h

centiseconds .byte 00h
```

;-----

RESET mov.w #__STACK_END,SP ; Initialize stackpointer

StopWDT mov.w #WDTPW|WDTHOLD,&WDTCTL ; Stop watchdog timer

;-----

; Main loop here

;-----

```
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
                         clr &TAIFG
                         clr &CCIFG
                         eint ; enable interrupts
```

```
Setup                    bis.b #11111111b, &P1DIR
                         bis.b #00001111b, &P2DIR

                         mov.w #0000001000010000b, &TA0CTL
                         mov.w #0000000000010000b, &TA0CCTL0
                         mov.w #0010011100010000b, &TA0CCR0

                         ;mov.w seconds,R5
                         ;mov.w centiseconds,R6

                         mov.w #array,R5
                         mov.w #array,R6
```

BLG 351E Microcomputer Laboratory – Experiment Report

Mainloop	<pre>mov.w #00001000b,&P2OUT mov.w @R6,&P1OUT clr &P1OUT mov.w #00000001b,&P2OUT mov.w @R5,&P1OUT clr &P1OUT jmp Mainloop</pre>
ISR	<pre>dint xor.w #0000000000010010b,&TA0CTL clr &P2IFG clr &TAIFG clr &CCIFG eint reti</pre>
TISR	<pre>dint inc.b R6 clr &TAIFG clr &CCIFG cmp.b #64h,R6 jnz cont mov.b #00h,R6 inc.b R5 clr &TAIFG clr &CCIFG</pre>
cont	<pre>eint reti</pre>

```
array    .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"
```

```
    .short ISR
```

```
    .sect ".int09"
```

```
    .short TISR
```

Firstly we store second and centisecond values on RAM in the data section. Then, we initialize the interrupt on PORT2, 6th button for stop and start interrupt and we clean the required flags. We assign the values to TA0CTL, TA0CCTL0 and TA0CCR0 registers bit by bit according to the MSP430 User Guide – Timer A chapter pages from 370 to 373. Address of the first element on the array is assigned to the R5 and R6 registers. Array is constructed as previous experiment to show the numbers from 0 to 9 on the 7 segment display. At the ISR part, if sixth button is pressed interrupt become enable and stop or start chronometer according to the toggle of the value on the TA0CTL register with 16 sized bit value which is the determined by us according to the document. In TISR part, we check if the centimeter part (rightest two part of 7-segment panel) access to the 100 (64h) or not. If it access to 100, set 0 to this part and increment the second part (leftest two part of 7-segment panel). Otherwise it continues the increment the centisecond. In Main part, we try to show the simply the first element of the arrays on the 7-segment panels, yet program is never enter the TISR interrupt. We did not find our error on this part. We wasted too much time on this problem and could not the write the BSD part.

3 CONCLUSION

We learn much about TIMER A and TISR interrupt yet, we cannot accomplish to the finish the experiment due to the not entering the this interrupt unfortunately. We wasted too much time on this problem and could not the write the BSD part.