Department of Computer Engineering

BLG 351E
Microcomputer Laboratory
Experiment Report

Experiment No : 2

Experiment Date : 17.10.2016

Group Number : Monday - 13

Group Members :

ID	Name	Surname
150130032	Baran	Kaya
40130051	Halil İbrahim	Onuş
150130002	Ahmet Seha	Çelenk
Click here to enter text.	Click here to enter text.	Click here to enter text.

Laboratory Assistant : Gökhan Seçinti

1 INTRODUCTION

In this experiment, we have learned how to use both P1 and P2 ports and how to use counter within registers.

2 COMPOSER STUDIO AND MSP430&KIT.EXPERIMENT

This experiment is consisting of three parts. In the first part, we made switch button for one of the leds on the board. In the second part, we used one register to keep a counter to count how many times pressed the button. In the final part, we used another button to clear the counter in register.

2.1 FIRST PART

Code:

```
SetupP1     bis.b     #0FFh,&P1DIR
            bit.b     #000h,&P2DIR

ReadS       bit.b     #004h,&P2IN
            jnz       ON
            jmp       ReadS

ON          bit.b     #004h,&P2IN
            jnz       ON
            xor.b     #005h,&P1OUT
            jmp       ReadS
```

In the setup part, bis.b sets P1 port for the output (LEDs) and bit.b sets P2 port for the input (button).

At the Reads loop, bit.b checks the 3rd button (0010 0000) for any input. If there is any input in the 3rd button, program jumping the ON loop with jnz (jump if not zero) command. But if there is no input program jump ReadS loop again and again until anybody push the button.

If there is an input, program jumps the ON loop and in this loop bit.b checks again the button for any input. If button is active the program jumps ON loop again and check the button until there is no input (Holding the button). However, if input become a zero, the program do not jump the ON loop. After that program use xor.b command for value $\oplus$ P1's #005h value (if our value is 1 it become 0, if our value is 0 it become 1) then jump ReadS again. In ReadS loop, program checks whether there is an input or not. If there is any input, program jumps ON loop. In this loop the LED is on and if input is still 1 program stays the ON loop, but if input is 0 then program jumps the ReadS again.

ReadS: Input = 0 → check input → Input = 0 → check input → Input = 1 → Jump ON

ON: Input = 1 → check input → Input = 1 → check input → Input = 0 → ON/OFF LED & jump ReadS

2.2 SECOND PART

Code:

```
            clr.w       R14
SetupP1     bis.b       #0FFh,&P1DIR
            bit.b       #000h,&P2DIR

ReadS       bit.b       #004h,&P2IN
            jnz         ON
            jmp         ReadS

ON          bit.b       #004h,&P2IN
            jnz         ON
            inc.b       R14
            mov.b       R14,&P1OUT
            ;xor.b      #005h,&P1OUT
            jmp         ReadS
```

First clr.w clears the inside of the register R14 then setup part sets input and output ports.

ReadS loop is the same as in the first part's ReadS loop.

In the ON loop, bit.b and jnz commands check the button for any input (Holding the button). If input is zero (that means we release the button) inc.b increase the value in the R14 register (it was 0 at the beginning) and then mov.b command shows the value inside the R14 register on the P1 port's leds (Binary format). After that xor.b checks the value and if it is 1 it become 0, if value is 0 it become 1. Finally, with jmp command program jumps to the ReadS loop again.

ReadS: Input = 0 → check input → Input = 0 → check input → Input = 1 → Jump ON

ON: Input = 1 → check input → Input = 1 → check input → Input = 0 → Increase R14 → Show it on LED → ON/OFF LED & jump ReadS

2.3 THIRD PART

Code:

```
            clr.w       R14
SetupP1     bis.b       #0FFh,&P1DIR
            bit.b       #000h,&P2DIR
```

```
ReadS   bit.b   #008h,&P2IN
        jnz     RESE
        bit.b   #004h,&P2IN
        jnz     ON
        jmp     ReadS
ON      bit.b   #004h,&P2IN
        jnz     ON
        inc.b   R14
        mov.b   R14,&P1OUT
        ;xor.b  #005h,&P1OUT
        jmp     ReadS

RESE    clr.b   R14
        mov.b   R14,&P1OUT
        jmp     ReadS
```

Setup part is the same.

In the ReadS part, program checks 4th button for any input. If there is an input on that button program jumps RESE part and that part resets the counter in the R14 register. However, if there is no input at 4th button program runs same as second part.

ON loop is the same as second part's ON loop.

In the RESE part, clr.b clears the inside of the R14 register and then show it on the P1 port's leds. After that program jumps ReadS loop and checks inputs again.

ReadS: 4th button = 0 → 3rd button = 0 → check buttons → 4th button = 0 → 3rd button = 1 → Jmp ON

ReadS: 4th button = 0 → 3rd button = 0 → check buttons → 4th button = 1 → Jump RESE

ON: Input = 1 → check input → Input = 1 → check input → Input = 0 → Increase R14 → Show it on LED → ON/OFF LED & jump ReadS

RESE: Clear R14 → Show it on LED → Jump ReadS

3 CONCLUSION

In this experiment, we improved our assembly language skills. We faced a little button problem that our first button was not working correctly. We pressed it several times and it did not work. We spent 1.5 hours to find it. After that we changed our code for using another button (3rd button) and then it started to work correctly.