

# Lab Sheet 4 – Online

## PIC16F84A Input/Output & C Language & Interrupts

### 1. Overview

In the previous three weeks, we have studied the instructions set of the PIC16F84A. The set is divided into:

1. Byte-oriented instructions that operate on bytes of data and further divided into:
  - a. Data transfer instruction,
  - b. Mathematical and logical instructions
2. Bit-oriented instructions that operate on a single bit.
3. Literal and control instructions.

In this lab we'll explore coding I/O ports for the PIC16F84A. Furthermore, we'll study the concept of interrupts. Four sources of interrupts are available in the PIC16F84A microcontroller:

1. External interrupt (RB0) (studied in this lab).
2. Interrupt on change (RB4-RB7) (studied in this lab).
3. Timer0 interrupt.
4. EEPROM write complete interrupt.

## 2. Simple Output Program

**Program 1:** First of all, create a project for this lab as was shown in the first week. Write the following program in the source code file:

```
        BSF        STATUS, RP0        ; Go to bank 1
        CLRF       TRISB              ; Setup all pins of PORTB as outputs
        BCF        STATUS, RP0        ; Go to bank 0
        MOVLW      0xFF               ; W = 0xFF
        MOVWF      PORTB              ; PORTB = W
LOOP    CALL        delay500ms         ; Call a subroutine resulting in a 0.5s delay
        COMF       PORTB              ; Complement all pins of PORTB
        GOTO       LOOP
```

**Code 1.** Program 1 Code

Before building your code, you should do the following:

- Download the file “Delay.inc” from Moodle to your project folder.
- Add the line (#include “Delay.inc”) just above the “main” label.
- What does the program do?

**Program 2:** Trace the following C program with your knowledge of the PIC16F84A keywords.

```
#include <htc.h>
#define _XTAL_FREQ 4000000
void main()
{
    TRISB=0X00;
    PORTB=0X00;
    while(1)
    {
        PORTB =~PORTB;
        __delay_ms(500);
    }
}
```

**Code 2.** Program 2 Code

- What does the program do?

**Exercise 1:** Implement the following program's functionality using the C language such that it only prints the numbers from 0 to 15 before going back to 0? Use the following program as an example. The example C code makes PORTB blink.

```

        BSF      STATUS, RP0
        CLRF     TRISB
        BCF      STATUS, RP0
        CLRF     PORTB
LOOP    CALL     delay500ms
        INCF     PORTB
        GOTO    LOOP

```

### 3. Interrupts

**Program 3:** Write the following program in the source code file (**Note that this program requires the Delay.inc include file**):

```

        BSF      STATUS, RP0
        CLRF     TRISA
        BSF      TRISB, 0
        BSF      INTCON, GIE      ; Enable Interrupts in General
        BSF      INTCON, INTE     ; Enable External Interrupt
        BCF      STATUS, RP0
LOOP    CLRF     PORTA
        GOTO    LOOP

ISR     BCF      INTCON, INTF     ; Clear External Interrupt Flag
        COMF     PORTA
        CALL     delay500ms
        RETFIE

```

Code 3 – Program 3

Please make sure to replace reftie with GOTO ISR at the top of your code before the main.

**Answer the following:**

- What happens if you replace the “RETFIE” instruction with “RETURN”? **Please justify your answer.** (no need to write a full program)
- What happens if you remove the “BCF INTCON, INTF”? In other words, what happens if you forget to clear the INTF? **Please justify your answer.** (no need to write a full program).

**4. Report Exercises**

1. Write a program that accepts two input signals, through switches, on pins RA0 and RA1. The program produces an output on PORTB according to the following table  
(*Hint: Checking two switches could be done using lookup tables or using multiple if statements*).

RA0	RA1	PORTB
0	0	0x00
1	0	0x0F
0	1	0xF0
1	1	0xFF

**Table 1** Truth table for exercise report exercise 1.

2. Write a program that performs the following operation:
  - a. Store an array of 5 elements starting at address 0x0C.
  - b. Accept five input numbers on PORTA and use them to fill the array. The microcontroller knows that a new input is available at PORTA when a push button connected to RB0 is pressed (please use interrupts).
  - c. After the last input is accepted, the microcontroller calculates the summation of the elements in the array and displays it back on PORTB (RB1 - RB7).
  - d. The output displayed on PORTB should be blinking every 1 second.