University of Jordan
School of Engineering
Department of Mechatronics Engineering
Microprocessor and Microcontroller Laboratory
0908432
Exp. 7: Basic Programming

Objectives

1. To become familiar with the process of writing an assembly language program for the PIC.
2. To demonstrate different methods of handling the I/O process.
3. To demonstrate different methods of handling interrupts.
4. To use the debugging facility of the MPLAB IDE to test programs.

Pre-lab Preparation:

1. Read chapter 7 of the PIC16F84 data sheet.
2. Read Appendix 1 carefully.
3. Write the required assembly language programs **carefully** with proper documentation.

Procedure:

Write an assembly code which operates a PIC16F84A that control a bottle labeling and packing machine.

Machine Sequence conveyor

1. The bottles pass through a conveyer belt, when the photocell sensor detects a bottle (External Interrupt), the label actuator(Solenoid) starts working and stops after 1.2 Sec.
2. 7-Segments show the numbers of labeled bottles, When the number of labeled bottles passing through the photocell sensor reaches nine the conveyer belt motor stops.
3. Nine bottles are packed into a cartoon. The cycle starts again when packing process is finished (Packing process needs to 3.6 Sec. to finished by Electro-Mechanical mechanism).
4. When the number of packed bottles reaches nine (Show the numbers of packed bottles on another digit of 7-Segments) the conveyer belt motor stops until the START pushbutton is pressed to start the cycle again.

```
;*******************************************************

; This program control a bottle labeling and packing machine.
; Photocell sensor is connected into RB0
; 7-Segments is connected to PORTB (We connect RB1 to a, RB2 to b …….And RB7 to g)
; Digits selection of bottles number 7-Segments is connected to RA0
; Digits selection of cartoon number 7-Segments is connected to RA1
; Conveyer belt motor is connected to RA2
; Label actuator is connected to RA3
; START pushbutton is connected to RA4
; The program uses a PIC16F84A running at crystal oscillator of frequency 4MHz.
;*******************************************************
 include "p16f84A.inc"
;*******************************************************
; Macro definitions

push    macro

        movwf       WTemp           ; WTemp must be reserved in all banks
        swapf       STATUS,W        ; store in W without affecting status bits
        banksel     StatusTemp      ; select StatusTemp bank
        movwf       StatusTemp      ; save STATUS
        endm

pop     macro

        banksel     StatusTemp      ; point to StatusTemp bank
        swapf       StatusTemp,W    ; unswap STATUS nibbles into W
        movwf       STATUS          ; restore STATUS
        swapf       WTemp,F         ; unswap W nibbles
        swapf       WTemp,W         ; restore W without affecting STATUS
        endm
;*******************************************************
; User-defined variables
        cblock      0x0C            ; bank 0 assignments
        WTemp
        StatusTemp
        ;……                        Add all variables here.
        endc
;*******************************************************
; Start of executable code
        org     0x00            ;Reset vector
        nop
        goto    Main
        org     0x04            ;
        goto            INT_SVC
```

```
        ;;;;;;; Main program ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
Main
        call    Initial         ;Initialize everything
MainLoop
        call    Bottle_Number               ;Check if the number of bottles reaches to nine
        call    Caroon_Number               ;Check if the number of packing bottles reaches to nine.
        goto    MainLoop                    ;Do it again
;;;;;;;   Initial subroutine ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; This subroutine performs all initializations of variables and registers.
Initial
        Return
;;;;;;; Bottle_Number subroutine ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; This subroutine Test if the number of bottles reaches to nine.
Bottle_Number
        Return
;*************************************************************
;;;;;;; Caroon_Number subroutine ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; This subroutine Test if the number of packing bottles reaches to nine.
Caroon_Number
        Return
;*************************************************************
;;;;;;; Delay subroutine ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; This subroutine to get a delay with 1.2 Sec.
Delay
        Return
;*************************************************************
; Bottle_Labeling
Bottle_Labeling
        bcf     INTCON, INTF    ;Clear the External interrupt flag
;       write the code here
        goto            POLL            ;Check for another interrupt
;*************************************************************
INT_SVC
push
POLL
        btfsc           INTCON, INTF        ; Check for an External Interrupt
        goto            Bottle_Labeling
;       btfsc   ...                         ; Check for another interrupt
;       call    ...
;       btfsc   ...                         ; Check for another interrupt
;       call    ...
        pop
        retfie

;*************************************************************
        End
```

General Guidelines for Writing your Programs

a. Always start with visualizing in your mind the process that should take place in the hardware.
b. Determine the inputs and outputs for the hardware.
c. Assign PIC ports to the hardware I/O.
d. Remember and always keep in mind the data flow cycle.
e. Never start writing code immediately in MPLAB, it wastes time and will very rarely give you what you want.
f. Start always with a flowchart on paper keeping in mind the above points. Try at first a general flowchart and then attempt to expand the flowchart into more detail to reflect the requirements of the program.
g. If you have done the above properly you will find that the flowchart will divide the program naturally into parts. You should now be able to write the code for each part.
h. Writing assembly takes time and needs patience, so be patient and careful with your code.
i. Write your comments to the code immediately with the code.
j. Study the programs that you have used in other experiments for writing style, hints and ideas.
k. Use the simulator in MPLAB to simulate your program.
l. You should demonstrate your working programs on the board to the lab supervisor.