Laborator 5. Microcontrolere – aplicatii

In cele doua exemple de mai jos este exemplificat modul de utilizare al porturilor de intrare si de iesire.

Construiti un proiect nou . Selectati ca device : PIC 16F84A. Creati un nou fisier, salvati-l cu numele counter.asm . Copiati codul de mai jos in acest fisier, adaugati-l in proiectul creat si construiti proiecutul. Folositi fereastra Watch pentru a urmari modificarile portului PORTB si variabilele J si K.

Counter.asm

```
        ;File Counter.asm
        ;Assembly code for PIC 16F84A microcontroller.

        ;Blinks LEDs on PORTB outputs in a sequential pattern.

        ;CPU configuration
        ;           PIC 16F84A, RC oscillator, watchdog timer off, power-up timer enabled.
        ;           Note: two underscore characters before the config command.

                    processor 16f84A
                    include  <p16f84A.inc>
                    __config _RC_OSC &_WDT_OFF &_PWRTE_ON

        ;Declare variables at two memory locations

        J           equ       H'1F'        ;J stored at hex address 1F.
        K           equ       H'1E'        ;K stored at hex address 1E.

        ;Program

                    org        0           ;Start program at address zero.

                    ;Set PORTB as output and initialize it.

                    movlw   B'00000000'    ;Move 8 binary zeros to the W (working) register.
                    tris    PORTB          ;Move Contents of W register to PORTB control register
                    movlw   B'00000001'    ;Move binary one to the W register.
                    movwf   PORTB          ;Move binary one to PORTB.

        mloop:      incfsz  PORTB,1        ;Increment PORTB bits

                    ;Waste some time by executing nested loops.

                    movlw   D'5'          ;Move decimal 5 to the W register.
                    movwf   J              ;Copy the decimal 5 from the W register to J.
        jloop:      movwf   K              ;Copy the decimal 5 from the W register to K.

        kloop:      decfsz   K,1           ;K=K-1, skip next instruction if zero.
                    goto     kloop
                    decfsz   J,1           ;J=J-1, skip next instruction if zero.
                    goto     jloop

                    ;Do it all again.

                    goto      mloop

                    end
```

In exemplul de mai jos se incrementeaza sau se decrementeaza valoarea portului PORTB in functie de intrarea portului PORTA astfel : daca bitul 0 de la PORTA (adica RA0) este 1 atunci PORTB este incrementat, daca bitul 1 de la PORTA (adica RA1) este 1 atunci PORTB este decrementat.

Folosind fereastra watch urmariti schimbarile din PORTA si PORTB. Pentru a simula rularea programului este necesara initializarea pinilor RA0 si RA1 astfel : meniu Debugger -> Stimulus -> New WorkBook->Asynch. In dreptul coloanei PIN/SRF se selecteaza RA0, respective RA1 pe randul doi. In dreptul coloanei Action se selecteaza Toggle pe amble randuri. Apoi, in timpul rularii in modul Animation al programului se apasa butonul din dreptul RA0 pe coloana Fire pentru incrementare si butonul din dreptul RA1 pe coloana Fire pentru decrementare

IncDec.asm

```
;File incdec.asm - Blinks LEDs on PORTB outputs in a sequential pattern.
;Configuration - RC oscillator, watchdog timer off, power-up timer enabled.
            processor 16f84A
            include  <p16f84A.inc>
            __config _RC_OSC &_WDT_OFF &_PWRTE_ON

;Declare variables at two memory locations
J           equ       H'1F'          ;J stored at hex address 1F.
K           equ       H'1E'          ;K stored at hex address 1E.

;Program
            org       0              ;Start program at address zero.

;Set PORTB as output and initialize it.

            movlw     B'00000000'    ;Move 8 binary zeros to the W (working) register
            tris      PORTB          ;Sets all of the pins of PORTB as outputs

            movlw     B'00011'
            tris      PORTA          ;Sets all of the pins of PORTA

            movlw     B'00000000'
            movwf     PORTB          ;Turns off all of the outputs of PORTB

;Scan input switches.
SCAN        btfsc         PORTA,0           ;Skip INCREMENT if PORTA pin 0 is low
                call       INCREMENT
            btfsc         PORTA,0           ;Skip WAIT if PORTA pin 0 is low
                call       WAIT

            btfsc     PORTA,1          ;Skip DECREMENT if PORTA pin 1 is low
                call       DECREMENT
            btfsc     PORTA,1          ;Skip WAIT if PORTA pin 1 is low
                call       WAIT

            goto      SCAN                    ;Check switches again.

;Increment output
INCREMENT   incfsz    PORTB
            btfss        PORTB,3        ;Skip return if PORTB pin 3 is high
            return
            btfss        PORTB,1        ;Skip return if PORTB pin 1 is high
            return
            movlw     B'00000000'    ;The number 10 to 15 was found so reset to 0
            movwf     PORTB          ;Turns off all of the outputs of PORTB
            return

;Decrement output
DECREMENT   decfsz    PORTB
            btfss        PORTB,3        ;Skip return if PORTB pin 3 is high
            return
            btfss        PORTB,2        ;Skip return if PORTB pin 2 is high
```

```
            return
            btfss       PORTB,1        ;Skip return if PORTB pin 1 is high
            return
            btfss       PORTB,0        ;Skip return if PORTB pin 0 is high
            return
            movlw     B'00001001'    ;The number 15 was found so reset to 9
            movwf     PORTB          ;Turns off all of the outputs of PORTB
            return


;Wait until the button is released (we want to count button presses)
WAIT        call      DELAY          ;Debounce
LOOK        btfsc     PORTA,0        ;Skip if PORTA pin 0 is low
            goto      LOOK
            btfsc     PORTA,1        ;Skip if PORTA pin 1 is low
            goto      LOOK
            return

;Delays for debounce
DELAY       movlw     D'1'           ;Time delay subroutine
            movwf     J
jloop       movwf     K
kloop       decfsz    K,1
            goto      kloop
            decfsz    J,1
            goto      jloop
            return

            end
```