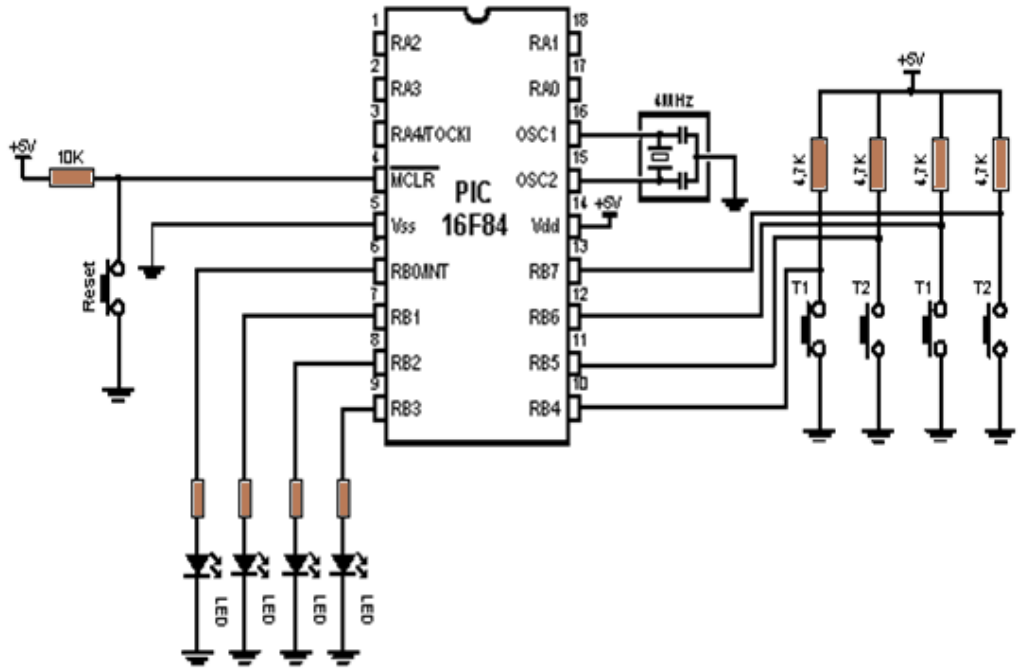
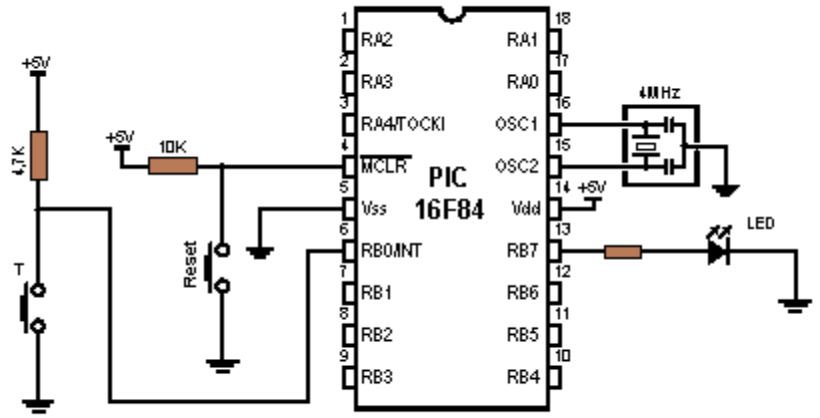
Interruption sur broches RB4-RB7



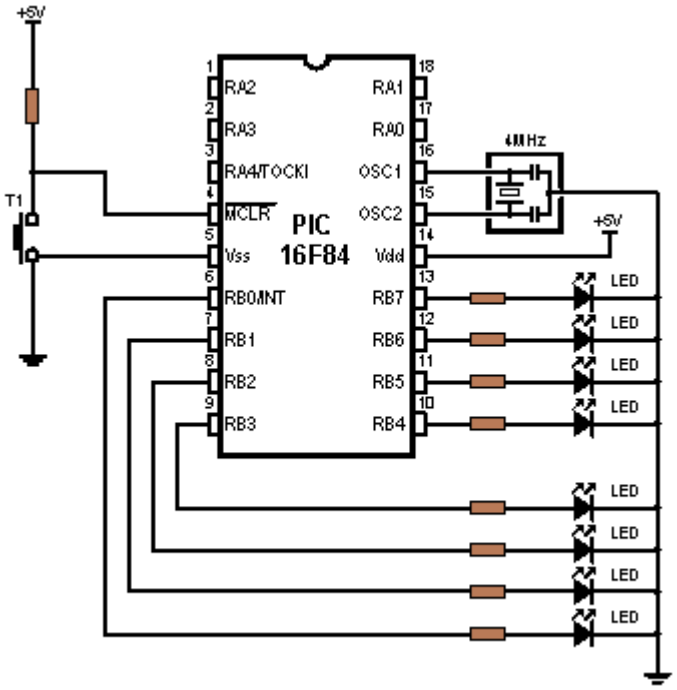
```
;***** Declaring and configuring a microcontroller *****
      PROCESSOR 16f84
      #include "p16f84.inc"
      __CONFIG _CP_OFF & _WDT_OFF & _PWRTE_ON & _XT_OSC

;***** Structure of program memory *****
      org 0x00
      goto Main
      org 0x04
      goto ISR
Main
      bsf 3,5
      movlw 0xf0          ;Higher four LED diodes are on
      movwf TRISB
      bcf 3,5
      movlw 0xff
      movwf PORTB
      bsf INTCON,RBIE     ;interrupt upon pin change enabled
      bsf INTCON,GIE      ;all interrupts are enabled

 Loop
       goto Loop           ;Main loop
 ISR
       bcf INTCON,RBIF     ;Clears the flag that indicates RB interrupt
                           ;took place thus enabling detection of
                           ;new interrupts in main program
       btfss PORTB,7       ;Determining which button caused the interrupt
       goto Led0
       btfss PORTB,6
       goto Led1
       btfss PORTB,5
       goto Led2
       btfss PORTB,4
       goto Led3
       retfie
 Led0
       bcf PORTB,0         ;Switch off diode LD0
       retfie
 Led1
       bcf PORTB,1         ;Switch off diode LD1
       retfie
 Led2
       bcf PORTB,2         ;Switch off diode LD2
       retfie
 Led3
       bcf PORTB,3         ;Switch off diode LD3
       retfie
__     End
```

Interruption sur la broche RB0



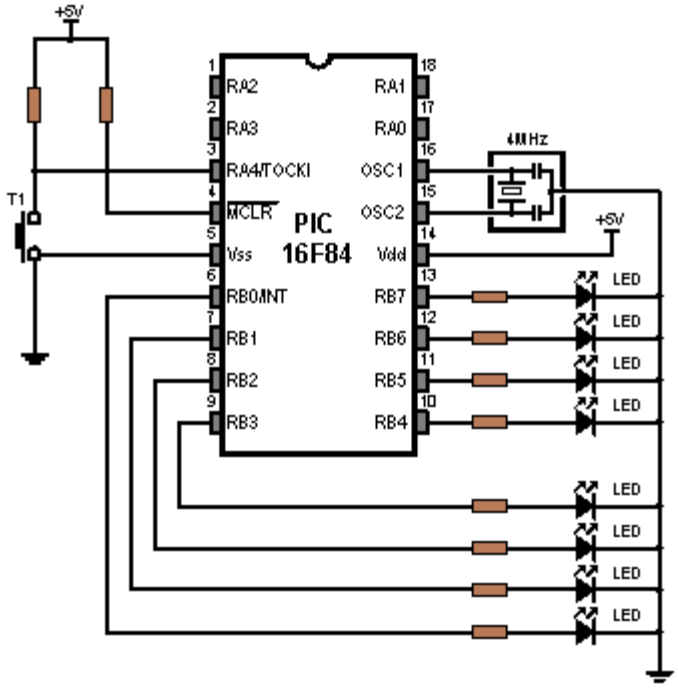

```
;***** Declaring and configuring a microcontroller *****
      PROCESSOR 16f84
      #include "p16f84.inc"
      __CONFIG _CP_OFF & _WDT_OFF & _PWRTE_ON & _XT_OSC

;***** Structure of program memory *****
      org 0x00
      goto Main
      org 0x04
      goto ISR
Main
      bsf 3,5
      movlw b'00000001'          ;RB0 is input, the rest are output
      movwf TRISB
      bcf OPTION_REG,INTEDG      ;interrupt occurs at faling edge
      bsf OPTION_REG,NOT_RBPU    ;internal pull-up resistors are off
      bcf 3,5
      clrf PORTB
      bsf PORTB,7                ;Only LED diode PORTB,7 is on
      bsf INTCON,INTE            ;interrupt RB0 enabled
      bsf INTCON,GIE             ;all interrupts enabled
Loop
      goto Loop                  ;Main loop

ISR
      bcf INTCON,INTF      ;Clears the flag that indicates RB interrupt
                           ;took place thus enabling detection of
                           ;new interrupts in main program
      btfss PORTB,7        ;Is LED7 on?
      goto Labl
      bcf PORTB,7          ;If true switch off LED7
      retfie
Labl
      bsf PORTB,7          ;If false switch on LED7
      retfie
      End
```

Interruption sur débordement du Timer TMR0



```
;***** Declaring and configuring a microcontroller *****
        PROCESSOR 16f84
        #include "p16f84.inc"
      __CONFIG _CP_OFF & _WDT_OFF & _PWRTE_ON & _XT_OSC
;***** Declaring variables *****
      cnt     equ 0x0c
;***** Structure of program memory *****
      ORG     0x00        ;Reset vector
      goto    Main
      ORG     0x04        ;Interrupt vector
      goto   ISR
Main
      bsf 3,5
      clrf    TRISB       ;Port B is output
      movlw .255
      movwf  TRISA        ;Port A is input
      movlw   B'10000100' ;Set prescaler to TMR0
      movwf   OPTION_REG  ;ps = 32=> TMR0 is incremented every 32us
      bcf 3,5
      clrf   PORTB        ;All the diodes are off by default
      bsf    INTCON,TOIE  ;Enable TMR0 interrupt
      movlw   .96         ;Initialize TMR0
                          ;Overflow occurs  every (255-96)*32us=5.088ms
      movwf   TMR0        ;Start the counter
      bsf INTCON,GIE      ;Interrupts are globally enabled
      clrf cnt
loop
      goto  loop          ;Remain at this line
ISR
      movlw   .96         ;Initialize TMR0 to ensure next interrupt
                          ;in 5ms
      movwf   TMR0
      bcf     INTCON,TOIF ;clear int. flag
      incf cnt,F
      movlw .196          ;Has one second elapsed?(1s~196*5.088 ms)
      subwf cnt,W
      btfss STATUS,Z
      retfie
      comf   PORTB,f      ;If true, complement the values of port B
      clrf cnt            ;and set the initial value of variable cnt
      retfie              ;If false, exit the interrupt routine
      End                 ;End of program
```

Interruption sur débordement du TMR0 connecté à une entrée externe (TOCKI)




```
;***** Declaring and configuring a microcontroller *****
        PROCESSOR 16f84
        #include "p16f84.inc"
      __CONFIG _CP_OFF & _WDT_OFF & _PWRTE_ON & _XT_OSC
;***** Declaring constants *****
      num_rev   equ .156 ;
;***** Structure of program memory *****
      ORG      0x00              ; Reset vector
      goto     Main
      ORG      0x04              ; Interrupt vector
      goto   ISR
Main  bsf 3,5
      clrf     TRISB             ; Port B is input
      movlw  B'111111'
      movwf  TRISA               ; Port A is output
      movlw  B'10100001'         ; External impulses on TOCKI increment TMR0
      movwf  OPTION_REG          ; ps = 4 TMR0 increments every 4 impulses
      bcf 3,5                    ; i.e. 1 revolution = 4 impulses
      clrf   PORTB               ; All diodes are off by default
      bsf    INTCON,TOIE         ; Enable TMR0 interrupt
      movlw  .256                ; Initialize TMR0 so that overflow
      sublw num_rev              ; occurs every 100 revolutions,
                                 ; i.e. TMR0 is set to 256-156=100
      movwf    TMR0              ; Start the counter
      bsf INTCON,GIE             ; Interrupts are globally enabled
```

```
loop
                        ; Display number of remaining revolutions
                        ; on port B
      movf TMR0,W
      sublw .256        ; number of remaining revolutions=256-TMR0
      movwf PORTB
      goto  loop        ; Remain at this line

ISR
      movlw   .256      ; Initialize TMR0 so it can count next
                        ; 100 revolutions
      sublw   broj_obrtaja
      movwf   TMR0

      bcf     INTCON,T0IF ; clear int. fleg
      retfie            ; return to main program

      End               ; End of program
```