



República Bolivariana de Venezuela  
Universidad Nacional Experimental Politécnica  
“Antonio José de Sucre”  
Vicerrectorado: Puerto Ordaz  
Asignatura: Microprocesadores I

**LABORATORIO 2 Y 3**  
**CRONÓMETRO DIGITAL**

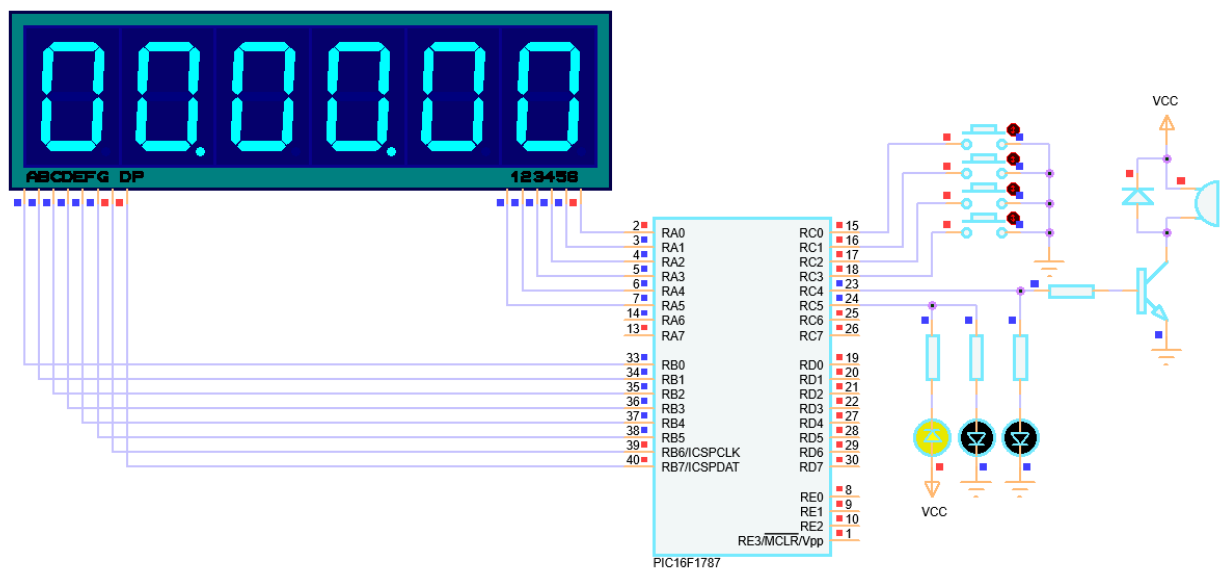
Docente:  
José Rodríguez

Alumno:  
Oliver Polo. C.I: 28.161.706

Ciudad Guayana, marzo de 2024

Se desea realizar el diseño de un cronómetro digital, el cual deberá mostrar los minutos, segundos y centésimas en display 7 segmentos. Dispondrá de pulsadores para las funciones de START/PAUSA y RESET. Adicionalmente el cronómetro tiene la posibilidad de operar en modo regresivo a partir de un valor ajustable por el usuario, cuando se agote el tiempo genera un beep intermitente de 10 segundos de duración o hasta que se presione cualquier botón. Dispondrá de 4 botones: 1- Menu/Enter 2- Arriba, 3- Abajo y 4- Atrás. Use estos botones para operar el diseño. Nota: observe el comportamiento de un cronómetro comercial y un timer regresivo. (El parámetro a ajustar deberá estar intermitente. (Ver un reloj digital comercial en modo ajuste)

### Configuración física:



**Imagen (1a). Diseño en Proteus (Cronómetro en Stand By)**

### Configuración de pines:

Los pines del RC0 al RC3 se usaron para entradas digitales (botones Pull Up), el pin RC4 se usó para activar la alarma indicando que se agotó en tiempo cuando está en modo Temporizador además cuenta con un indicador LED cian por el mismo pin y el pin

RC5 se usó para indicar el estado (modo Cronómetro LED amarillo, modo Temporizador LED rosado), los pines del RB0 al RB6 se usan como salidas digitales para enviar el número 7 segmentos al display y el pin RB7 se usó para encender los puntos de separación, los pines del RA0 al RA5 se usaron para hacer el barrido mientras que el pin RA6 se usó como bandera para corregir el barrido.

### **Funcionamiento:**

Al iniciar el dispositivo el modo principal es Cronómetro el cual queda en Stand By hasta que el usuario empiece a operarlo.

### **Botones:**

A: pin RC0

B: pin RC1

C: pin RC2

D: pin RC3

- Si se pulsa el botón A, el Cronómetro empieza a correr
- Si el cronómetro está corriendo y se pulsa el botón A, éste se pausa
- Si se pulsa el botón B y el dispositivo se encuentra en modo cronómetro, éste se reinicia (aunque esté corriendo o esté en pausa)
- El botón C no cumple ninguna función en modo Cronómetro
- Si el dispositivo está en modo Cronómetro y se pulsa el botón D, éste pasa a modo Temporizador
- Si el dispositivo está en modo Temporizador los botones A y D se usan para desplazar el cursor de ajuste a la izquierda o derecha respectivamente
- Si el dispositivo está en modo Temporizador los botones B y C se usan para incrementar o decrementar respectivamente el número en el que se encuentra el cursor

- Si el dispositivo está en modo Temporizador y el cursor se encuentra en las centésimas y se pulsa el botón D, éste pasa a modo Cronómetro
- Estando en modo Temporizador, si el cursor se encuentra en los minutos y ya el usuario ya ingreso el tiempo, si se pulsa el botón A empieza la temporización. En caso de que el usuario no haya ingresado el tiempo y se pulse el botón A, el cursor se regresa a las centésimas
- Si ya se está descontando el tiempo de la temporización y se pulsa el botón A, éste se pausa, al igual que si se encuentra en pausa y se pulsa el botón A, éste continúa descontando.
- Para reiniciar el Temporizador se pulsa el botón D

### Modo Cronómetro:

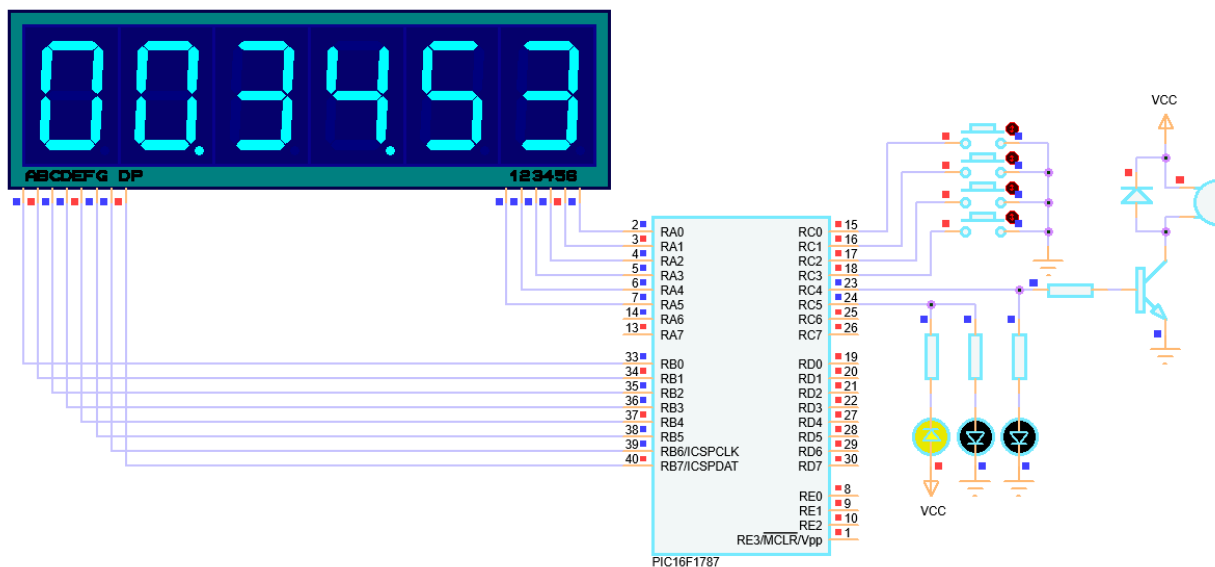


Imagen (2a): Modo Cronómetro

## Modo Temporizador:

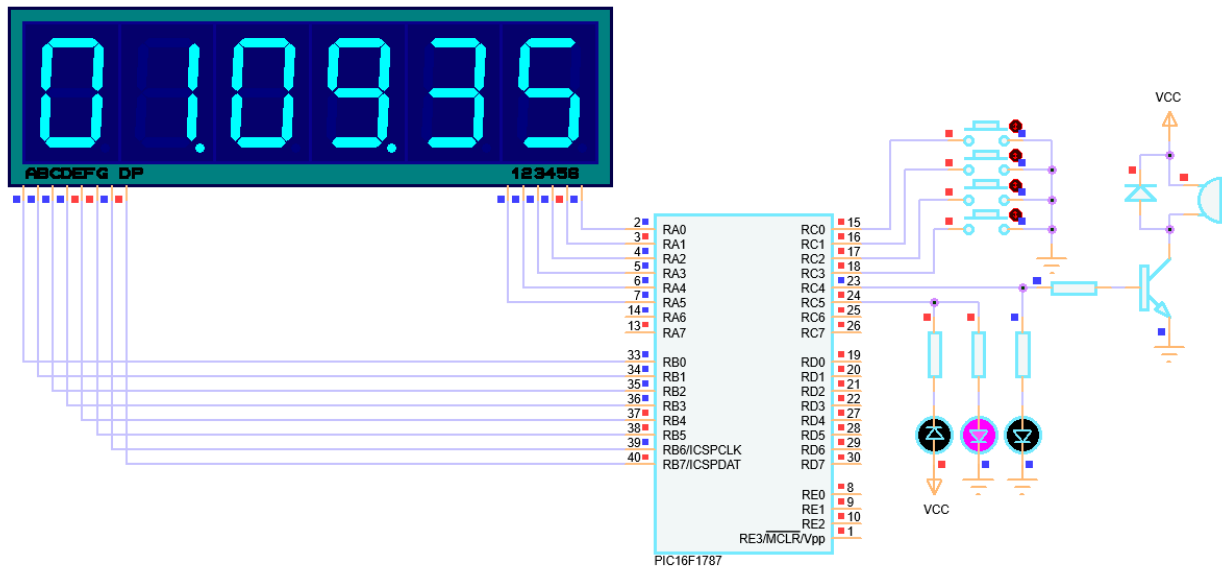
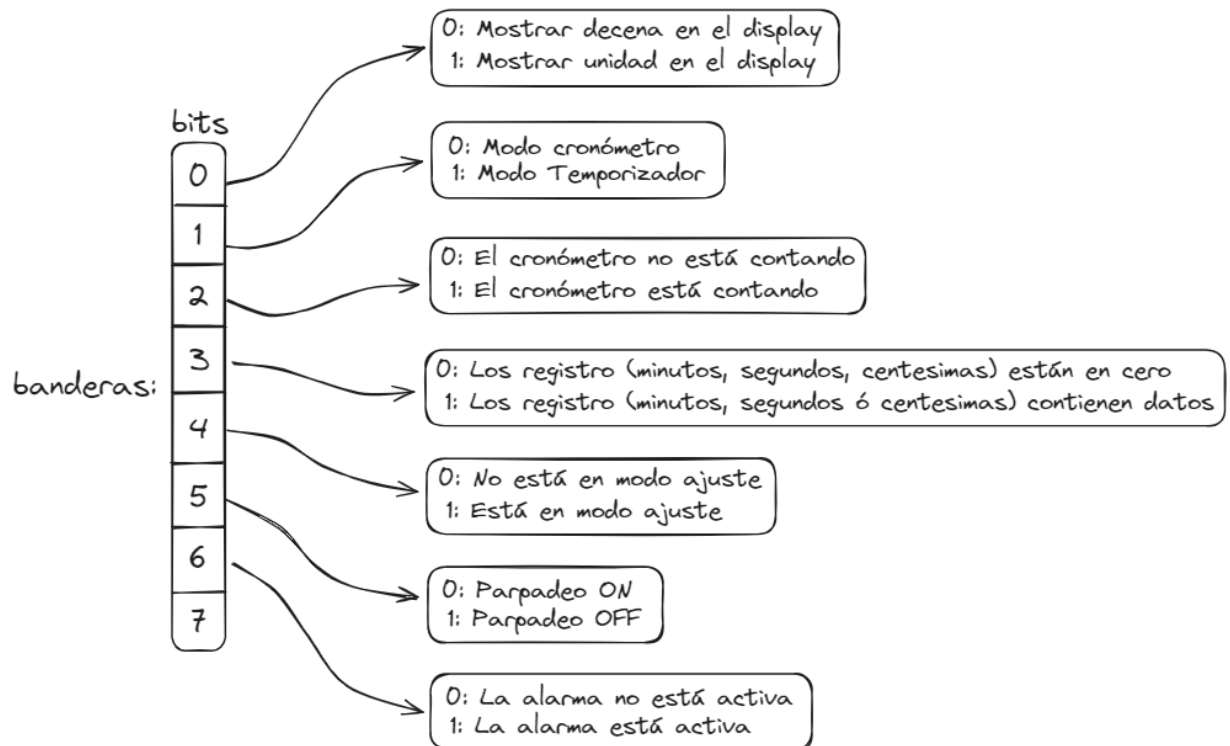


Imagen (3a): Modo Temporizador

## Banderas de control de flujo:



## Cálculo del TMR0

El tiempo de refrescamiento del display será 1ms, éste también será el tiempo base del TMR0.

$$\text{Temporización} = 4 * T_{osc}(256 - TMR0) * (\text{divisor de frecuencia})$$

Queriendo un refrescamiento de 1ms y sustituyendo, tomando en cuenta que la oscilación interna será 4Mhz, nos queda:

$$1000\mu s = 1\mu s(256 - TMR0) * (\text{divisor de frecuencia})$$

Despejando:

$$\frac{1000\mu s}{1\mu s * (\text{divisor de frecuencia})} = (256 - TMR0)$$

$$TMR0 = 256 - \frac{1000}{(\text{divisor de frecuencia})}$$

Tomando un divisor de frecuencia igual a 8:

$$TMR0 = 256 - \frac{1000}{8} = 256 - 125$$

$$TMR0 = 131$$

## Asignación de nombre a registros.

```
list p=16f1787
include <p16f1787.inc>

__CONFIG __CONFIG1, _FOSC_INTOSC & _WDTE_OFF

centesimas      equ      0x20
segundos        equ      0x21
minutos         equ      0x22
banderas        equ      0x23
barrido         equ      PORTA
current_display equ      0x24
bcd_unidad      equ      0x25
bcd_decena      equ      0x26
check_zero      equ      0x27 ;Cuando se decrementa, si check_zero = .3 se detuvo el temp
cursor          equ      0x28 ;Almacena el registro del display que está seteando el usuario
cont_ms         equ      0x29
parp_ms         equ      0x30
alarma1_ms      equ      0x31
alarma2_ms      equ      0x32
alarma3_ms      equ      0x33
parp_display    equ      0x34
```

## Configuración para escribir el código en diferentes partes de la página.

```
;Inicio
org      0x00
goto     CONFIG__
;Interrupcion
org      0x04
goto     INTER
org      0x05
```

## Configuraciones de puertos, divisor de frecuencia y limpieza de registros

```
CONFIG__      ;Configuraciï¿½n de puertos y registros

banksel OSCCON
movlw 0x6F
movwf OSCCON
banksel ANSELA
clrf ANSELA
clrf ANSELB

;Configuraciï¿½n de entradas y salidas
banksel TRISA
movlw b'10000000'
movwf TRISA
clrf TRISB
movlw b'11001111'
movwf TRISC
banksel OPTION_REG
movlw b'00000010'
movwf OPTION_REG

;Limpiando los registros
clrf BSR
clrf centesimas
clrf segundos
clrf minutos
clrf banderas
clrf barrido
clrf FSR0L
clrf FSR0H
clrf FSR1L
clrf FSR1H
clrf PORTC
```



## Inicialización de registros

```
;Inicializando registro de barrido
movlw    0x01
movwf    barrido
movwf    PORTA

;Inicializando registro de display
movlw    0x20
movwf    current_display
movwf    cursor

;Inicializando registro para contar 10ms
movlw    .10
movwf    cont_ms

;Inicializando tiempo de parpadeo para modo ajuste (60ms)
movlw    .60
movwf    parp_ms

;Inicializando tiempo de alarma (50*2*100)ms = 10000ms = 10s
movlw    .50
movwf    alarma1_ms
movlw    .2
movwf    alarma2_ms
movlw    .100
movwf    alarma3_ms

;Configuración de interrupciones
movlw    b'11101000'
movwf    INTCON
banksel  IOCCN
movlw    0x0f
movwf    IOCCN
clrf     BSR
```

**Subrutinas de interrupción por desbordamiento del TMR0 y por Cambio de nivel en el puerto C.**

```
INTER      call    SHOW_STATUS
           btfsc   INTCON, TMR0IF
           goto    TMR_INT
           goto    FLANCO_INT

;Interrupción por TMR0
TMR_INT    call    ROTA_DISP
           call    SHOW_DISP

           btfsc   banderas, 6
           goto    ALARMA
           btfss   banderas, 1
           goto    CRONOMETRO
           btfsc   banderas, 3
           goto    TEMPORIZADOR

RESET_TMR  movlw   .131
           movwf   TMR0
           bcf     INTCON, TMR0IF
           retfie

;Interrupción por cambio de nivel (flanco de bajada)
FLANCO_INT call    CHECK_ALARMA
           banksel IOCCF
           btfsc   IOCCF, 0
           call    BOTON0
           btfsc   IOCCF, 1
           call    BOTON1
           btfsc   IOCCF, 2
           call    BOTON2
           btfsc   IOCCF, 3
           call    BOTON3
           banksel IOCCF
           clrf    IOCCF
           clrf    BSR
           retfie
```

## Subrutinas para el botón A (multipropósito).

```
;Configuraciï¿½n para el botï¿½n 0 (RC0)
BOTON0      clrf      BSR
            btfss     banderas, 1
            goto      BTN0_CRON
            goto      BTN0_TEMP

;Botï¿½n START/PAUSE
BTN0_CRON   btfsc     banderas, 2
            goto      PAUSE
            bsf       banderas, 2
            return

;Si el cronometro estï¿½ contando se detiene
PAUSE       bcf       banderas, 2
            bcf       IOCCF, 0
            return

BTN0_TEMP   btfss     banderas, 4
            goto      BTN0_TEMP_PAUSE
            call      RL_CURSOR
            banksel   IOCCF
            return

BTN0_TEMP_PAUSE
            btfsc     banderas, 2
            goto      TEMP_PAUSE
            bsf       banderas, 2
            banksel   IOCCF
            return

TEMP_PAUSE  bcf       banderas, 2
            banksel   IOCCF
            return

;
```

## Subrutinas para el botón B (multipropósito).

```
;Configuraciï¿½n para el botï¿½n 1 (RC1)
```

---

```
BOTON1      clrfs    BSR  
            btfss    banderas, 1  
            goto     BTN1_CRON  
            goto     BTN1_TEMP
```

```
BTN1_CRON    clrfs    BSR  
            bcf      banderas, 2  
            clrfs    centesimas  
            clrfs    segundos  
            clrfs    minutos  
            banksel  IOCCF  
            return
```

```
BTN1_TEMP    btfss    banderas, 4  
            goto     $+4  
            movf     cursor, W  
            movwf    FSR1L  
            ;incf    INDF1, F  
            call     INC_FILE  
            banksel  IOCCF  
            return
```

```
;
```

---

### Subrutina para el botón C.

```
;Configuraciï¿½n para el botï¿½n 2 (RC2)
BOTON2      clrfs    BSR
            btfss    banderas, 4
            goto     $+6
            btfss    banderas, 1
            goto     $+4
            movf     cursor, W
            movwf    FSR1L
            call     DEC_FILE
            banksel  IOCCF
            return

;
```

### Subrutinas para el botón D (multipropósito).

```
;Configuraciï¿½n para el botï¿½n 3 (RC3)
BOTON3      clrfs    BSR
            btfsc    banderas, 4
            goto     BTN3_ADJ
            bcf      banderas, 2
            clrfs    centesimas
            clrfs    segundos
            clrfs    minutos
            bsf      banderas, 1
            bcf      banderas, 2
            bsf      banderas, 4
            return

BTN3_ADJ    movlw    centesimas
            xorwf    cursor, W
            btfss    STATUS, Z
            goto     RR_CURSOR
            bcf      banderas, 4
            bcf      banderas, 1
            clrfs    centesimas
            clrfs    segundos
            clrfs    minutos
            banksel  IOCCF
            return

;
```

## Subrutina para incrementar los registros en modo ajuste.

```
;Incrementa el registro en modo ajuste
;Si el valor de las centesimas es 99, el incremento coloca el registro en 0
;Si el valor de los segundos o minutos es 59, el incremento coloca el registro en 0
INC_FILE      movlw    centesimas
              xorwf    cursor, W
              btfsc   STATUS, Z
              goto     INC_LIM99
              goto     INC_LIM59
INC_LIM99     incf     INDF1, F
              movf     INDF1, W
              xorlw    .100
              btfss   STATUS, Z
              return
              clrf     INDF1
              return
INC_LIM59     incf     INDF1, F
              movf     INDF1, W
              xorlw    .60
              btfss   STATUS, Z
              return
              clrf     INDF1
              return
```

## Subrutina para decrementar los registros en modo ajuste.

```
;Decrementa el registro en modo ajuste
;Si el valor de las centesimas es 0, el decremento coloca el registro en 99
;Si el valor de los segundos o minutos es 0, el decremento coloca el registro en 59
DEC_FILE      movlw    centesimas
              xorwf    cursor, W
              btfsc   STATUS, Z
              goto     DEC_LIM99
              goto     DEC_LIM59
DEC_LIM99     decf     INDF1, F
              movf     INDF1, W
              xorlw    .255
              btfss   STATUS, Z
              return
              movlw    .99
              movwf    INDF1
              return
DEC_LIM59     decf     INDF1, F
              movf     INDF1, W
              xorlw    .255
              btfss   STATUS, Z
              return
              movlw    .59
              movwf    INDF1
              return
```

## Subrutinas para rotar el cursor a la izquierda en modo ajuste.

```
;Rota el cursor de ajuste a la izquierda
RL_CURSOR    movlw    minutos
              xorwf    cursor, W
              btfsc    STATUS, Z
              goto     CHECK_RLCUR
              incf     cursor, F
              return

;Comprueba si los registros ya tienen datos
;sino el cursor vuelve al display de centesimas
CHECK_RLCUR  movlw    .0
              xorwf    centesimas, W
              btfss    STATUS, Z
              bsf      banderas, 3
              movlw    .0
              xorwf    segundos, W
              btfss    STATUS, Z
              bsf      banderas, 3
              movlw    .0
              xorwf    minutos, W
              btfss    STATUS, Z
              bsf      banderas, 3
              btfss    banderas, 3
              goto     $+6
              movf     minutos, W
              xorwf    cursor, W
              btfss    STATUS, Z
              goto     $+2
              goto     TEMPORIZADOR
              movlw    0x20
              movwf    cursor
              return
```

**Subrutinas para rotar el cursor a la derecha en modo ajuste.**

```
BTN3_ADJ    movlw    centesimas
             xorwf    cursor, W
             btfss    STATUS, Z
             goto     RR_CURSOR
             bcf       banderas, 4
             bcf       banderas, 1
             clrf      centesimas
             clrf      segundos
             clrf      minutos
             banksel   IOCCF
             return

RR_CURSOR    decf      cursor, F
             banksel   IOCCF
             return
```



## Subrutinas para mostrar los dígitos en el display.

```
;Mostrando los dígitos en los display (bcd_decena, bcd_unidad)
SHOW_DISP    movf    INDF0, W
              call    DECIMAL_A_BCD
              btfsc   banderas, 0
              goto    SHOW_UNI
              goto    SHOW_DEC
SHOW_UNI      movf    bcd_unidad, W
              call    TABLA_7SEG_AC
              btfsc   banderas, 4
              call    BCD_PARP
              movwf   PORTB
              call    PUNTOS
              bcf     banderas, 0
              return
SHOW_DEC      movf    bcd_decena, W
              call    TABLA_7SEG_AC
              btfsc   banderas, 4
              call    BCD_PARP
              movwf   PORTB
              call    PUNTOS
              bsf     banderas, 0
              return

;Activa el display a la izq y lo deja así por 1ms (base de tiempo del TMR0)
;barrido = PORTA
ROTA_DISP    rlf     barrido, F
              btfsc   barrido, 6
              goto    CORREGIR
              btfsc   banderas, 0
              incf    current_display, F
              movf    current_display, W
              movwf   FSR0L
              return
CORREGIR      movlw   0x01
              movwf   barrido
              movlw   0x20
              movwf   current_display
              movwf   FSR0L
              return
```

## Subrutinas para cronómetro.

```
CRONOMETRO    btfss    banderas, 2
               goto     RESET_TMR
               decfsz    cont_ms, F
               goto     RESET_TMR
               call      INCREMENTA
               goto     RESET_TMR

INCREMENTA     movlw     .10
               movwf     cont_ms
               movf       centesimas, W
               xorlw     .99
               btfsc     STATUS, Z
               goto      INC_SEG
               incf       centesimas, F
               return

INC_SEG        clrfs     centesimas
               movf       segundos, W
               xorlw     .59
               btfsc     STATUS, Z
               goto      INC_MIN
               incf       segundos, F
               return

INC_MIN        clrfs     segundos
               movf       minutos, W
               xorlw     .59
               btfsc     STATUS, Z
               goto      CLR_INC
               incf       minutos, F
               return

CLR_INC        clrfs     centesimas
               clrfs     segundos
               clrfs     minutos
               return
```

## Subrutinas para Temporizador.

```
TEMPORIZADOR
    btfss    banderas, 2
    decfsz   cont_ms, F
    goto     RESET_TMR
    call     DECREMENTA
    goto     RESET_TMR

DECREMENTA    bcf      banderas, 4
              goto     CHECK_DEC
DECR          movlw    .10
              movwf    cont_ms
              movf      centesimas, W
              xorlw     .0
              btfsc    STATUS, Z
              goto     DEC_SEG
              decf      centesimas, F
              return
DEC_SEG       movlw    .99
              movwf    centesimas
              movf      segundos, W
              xorlw     .0
              btfsc    STATUS, Z
              goto     DEC_MIN
              decf      segundos, F
              return
DEC_MIN       movlw    .59
              movwf    segundos
              decf      minutos, F
```

```
;Se verifica si todos los registros están en cero (minutos=0, segundos=0, centesimas=0)
CHECK_DEC     movf      minutos, W
              xorlw     .0
              btfss    STATUS, Z
              goto     DECR
              movf      segundos, W
              xorlw     .0
              btfss    STATUS, Z
              goto     DECR
              movf      centesimas, W
              xorlw     .0
              btfss    STATUS, Z
              goto     DECR
              bcf      banderas, 3
              bsf      banderas, 4
              movlw    centesimas
              movwf    cursor
              bsf      banderas, 6 ;Si todos los registros llegan a cero se activa la alarma
              retfie
```

## Subrutinas de alarma.

```
;Se hacen varias temporizaciones "anidadadas", cada dos temporizaciones de 100ms
;se alterna el bit 4 del puerto C, haciendo esto 50 veces la alarma dura 10s
ALARMA    decfsz   alarma3_ms, F
          goto     RESET_TMR
          movlw    .100
          movwf    alarma3_ms
          decfsz   alarma2_ms, F
          goto     ALARMA
          call     TOGGLE_BEEP
          movlw    .2
          movwf    alarma2_ms
          decfsz   alarma1_ms, F
          goto     ALARMA
          bcf      banderas, 6
          movlw    .50
          movwf    alarma1_ms
          bcf      PORTC, 4
          goto     RESET_TMR
TOGGLE_BEEP btfss   PORTC, 4
          goto     $+3
          bcf      PORTC, 4
          return
          bsf      PORTC, 4
          return

;Luego de la interrupci n por cambio de nivel, si est  transcurriendo el tiempo de alarma
;se detiene colocando a cero el bit 4 del puerto C
CHECK_ALARM
          btfss   banderas, 6
          return
          bcf      banderas, 6
          bcf      PORTC, 4
          goto     RESET_TMR
```

### Subrutina de parpadeo para modo ajuste.

```
;Si esti: en modo ajuste, envi: al display donde se encuentra el cursor
;0xff por 60ms y luego el dato que deberi: ir al display por otros 60ms
BCD_PARP      movwf    parp_display
              movf     FSR0L, W
              xorwf    cursor, W
              btfsc    STATUS, Z
              goto     COMP_PARP
              movf     parp_display, W
              return
COMP_PARP     decfsz   parp_ms, F
              goto     FIN_PARP
              movlw    .60
              movwf    parp_ms
              btfsc    banderas, 5
              goto     PARP_ON
              goto     PARP_OFF
PARP_ON       bcf      banderas, 5
              movf     parp_display, W
              return
PARP_OFF      bsf      banderas, 5
              movlw    0xff
              return
FIN_PARP      btfss    banderas, 5
              goto     $+3
              movlw    0xff
              return
              movf     parp_display, W
              return
```

### Subrutina para separar un número de dos dígitos en unidad y decena.

```
;Separa los digitos de un registro
;deja la decena en bcd_decena y la unidad en bcd_unidad
DECIMAL_A_BCD
              movwf    bcd_unidad
              clrf     bcd_decena
RESTA_DECENA  movlw    .10
              subwf    bcd_unidad
              btfss    STATUS, C
              goto     SUMA_UNIDAD
              incf     bcd_decena, F
              goto     RESTA_DECENA
SUMA_UNIDAD  addwf     bcd_unidad, F
              return
```

### Subrutina para mostrar el modo en el puerto C, Cronómetro o Temporizador.

```
;Muestra si está en modo Cronómetro o Temporizador a través del bit 5 del puerto C
SHOW_STATUS btfss  banderas, 1
             goto   $+3
             bsf     PORTC, 5
             return
             bcf     PORTC, 5
             return
```

### Subrutina para encender los puntos de separación.

(minutos . segundos . centesimas)

```
PUNTOS      btfss  banderas, 0
             return
             movf   current_display, W
             xorlw  centesimas
             btfsc  STATUS, Z
             return
             bcf    PORTB, 7
             return
```

### Subrutina para convertir un número a su equivalente en 7 segmentos.

```
;Tabla display 7seg i: nodo com:
TABLA_7SEG_AC
             brw
             retlw b'11000000' ; CERO
             retlw b'11111001' ; UNO
             retlw b'10100100' ; DOS
             retlw b'10110000' ; TRES
             retlw b'10011001' ; CUATRO
             retlw b'10010010' ; CINCO
             retlw b'10000010' ; SEIS
             retlw b'11111000' ; SIETE
             retlw b'10000000' ; OCHO
             retlw b'10010000' ; NUEVE
end
```

## Código Ensamblador Cronómetro/Temporizador

list p=16f1787

include <p16f1787.inc>

\_\_CONFIG \_\_CONFIG1, \_FOSC\_INTOSC & \_WDTE\_OFF

centesimas	equ	0x20
segundos	equ	0x21
minutos	equ	0x22
banderas	equ	0x23
barrido	equ	PORTA
current_display	equ	0x24
bcd_unidad	equ	0x25
bcd_decena	equ	0x26
check_zero	equ	0x27 ;Cuando se decrementa, si check_zero = .3 se detuvo el temp
cursor	equ	0x28 ;Almacena el registro del display que estí½ seteando el usuario
cont_ms	equ	0x29
parp_ms	equ	0x30
alarma1_ms	equ	0x31
alarma2_ms	equ	0x32
alarma3_ms	equ	0x33

parp\_display      equ          0x34

;Inicio

org    0x00

goto  CONFIG\_\_

;Interrupcion

org    0x04

goto  INTER

org    0x05

;Configuraci3n de puertos y registros

CONFIG\_\_

banksel OSCCON

movlw      0x6F

movwf      OSCCON

banksel ANSELA

clrf    ANSELA

clrf    ANSELB

;Configuraci3n de entradas y salidas

banksel TRISA

movlwb'10000000'

movwfTRISA

clrf    TRISB



```
movlwb'11001111'  
movwfTRISC  
banksel    OPTION_REG  
movlwb'00000010'  
movwfOPTION_REG
```

;Limpiando los registros

```
clrf    BSR  
clrf    centesimas  
clrf    segundos  
clrf    minutos  
clrf    banderas  
clrf    barrido  
clrf    FSR0L  
clrf    FSR0H  
clrf    FSR1L  
clrf    FSR1H  
clrf    PORTC
```

;Iniciando registro de barrido

```
movlw0x01  
movwfbarrido  
movwfPORTA
```

;Iniciando registro de display

movlw0x20

movwfcursor\_display

movwfcursor

;Iniciando registro para contar 10ms

movlw.10

movwfcont\_ms

;Iniciando tiempo de parpadeo para modo ajuste (60ms)

movlw.60

movwfparp\_ms

;Iniciando tiempo de alarma  $(50 \times 2 \times 100)\text{ms} = 10000\text{ms} = 10\text{s}$

movlw.50

movwfalarma1\_ms

movlw.2

movwfalarma2\_ms

movlw.100

movwfalarma3\_ms

;Configuraci3n de interrupciones

```

movlwb'11101000'

movwflINTCON

banksel    IOCCN

movlw0x0f

movwflIOCCN

clrf    BSR

```

```

PROGRAMAgoto $

```

```

INTER      call    SHOW_STATUS

           btfsc   INTCON, TMR0IF

           goto    TMR_INT

           goto    FLANCO_INT

```

;Interrupcii 1/2n por TMR0

```

TMR_INT      call    ROTA_DISP

           call    SHOW_DISP

           btfsc   banderas, 6

           goto    ALARMA

           btfss   banderas, 1

           goto    CRONOMETRO

```

```
btfsc banderas, 3
goto TEMPORIZADOR
```

```
RESET_TMR    movlw .131
              movwf TMR0
              bcf      INTCON, TMR0IF
              retfie
```

;Interrupci3n por cambio de nivel (flanco de bajada)

```
FLANCO_INT    call  CHECK_ALARMA
              banksel IOCCF
              btfsc  IOCCF, 0
              call  BOTON0
              btfsc  IOCCF, 1
              call  BOTON1
              btfsc  IOCCF, 2
              call  BOTON2
              btfsc  IOCCF, 3
              call  BOTON3
              banksel IOCCF
              clrf   IOCCF
              clrf   BSR
              retfie
```

;Configuraci3n para el bot3n 0  
(RC0)\_\_\_\_\_

BOTON0           clrf   BSR  
  
                  btfss banderas, 1  
  
                  goto  BTN0\_CRON  
  
                  goto  BTN0\_TEMP

;Bot3n START/PAUSE

BTN0\_CRON       btfsc banderas, 2  
  
                  goto  PAUSE  
  
                  bsf       banderas, 2  
  
                  return

;Si el cronometro est3 contando se detiene

PAUSE           bcf       banderas, 2  
  
                  bcf       IOCCF, 0  
  
                  return

BTN0\_TEMPbtfss banderas, 4  
  
                  goto  BTN0\_TEMP\_PAUSE  
  
                  call  RL\_CURSOR  
  
                  banksel   IOCCF

return

BTN0\_TEMP\_PAUSE

btfsc banderas, 2

goto TEMP\_PAUSE

bsf banderas, 2

banksel IOCCF

return

TEMP\_PAUSE bcf banderas, 2

banksel IOCCF

return

;  
\_\_\_\_\_  
\_\_\_\_\_

;Configuraci3n para el bot3n 1  
(RC1)\_\_\_\_\_

BOTON1 clrf BSR

btfss banderas, 1

goto BTN1\_CRON

goto BTN1\_TEMP

BTN1\_CRON clrf BSR

```

        bcf      banderas, 2

        clrf     centesimas

        clrf     segundos

        clrf     minutos

        banksel   IOCCF

        return

```

```

BTN1_TEMPbt fss banderas, 4

        goto    $+4

        movf    cursor, W

        movwf   FSR1L

        ;incf   INDF1, F

        call    INC_FILE

        banksel   IOCCF

        return

```

```

;
_____
_____

```

```

;Configuraci3n para el bot3n 2
(RC2)_____

```

```

BOTON2      clrf   BSR

              btfss banderas, 4

```

```

goto    $+6

btfss   banderas, 1

goto    $+4

movf    cursor, W

movwf   FSR1L

call    DEC_FILE

banksel    IOCCF

return

```

```

;_____
_____

```

```

;Configuraci3n para el bot3n 3
(RC3)_____

```

```

BOTON3      clrf    BSR

              btfsc   banderas, 4

              goto    BTN3_ADJ

              bcf      banderas, 2

              clrf     centesimas

              clrf     segundos

              clrf     minutos

              bsf       banderas, 1

              bcf       banderas, 2

```



```
bsf      banderas, 4
return
```

```
BTN3_ADJ  movlw centesimas
```

```
    xorwf cursor, W
    btfss STATUS, Z
    goto  RR_CURSOR
    bcf      banderas, 4
    bcf      banderas, 1
    clrf     centesimas
    clrf     segundos
    clrf     minutos
    banksel  IOCCF
    return
```

```
;
```

---

```
;Incrementa el registro en modo ajuste
```

```
;Si el valor de las centesimas es 99, el incremento coloca el registro en 0
```

```
;Si el valor de los segundos o minutos es 59, el incremento coloca el registro en 0
```

```
INC_FILE  movlw centesimas
```

```

        xorwf cursor, W
        btfsc STATUS, Z
        goto INC_LIM99
        goto INC_LIM59
INC_LIM99 incf INDF1, F
        movf INDF1, W
        xorlw .100
        btfss STATUS, Z
        return
        clrf INDF1
        return
INC_LIM59 incf INDF1, F
        movf INDF1, W
        xorlw .60
        btfss STATUS, Z
        return
        clrf INDF1
        return

```

;Decrementa el registro en modo ajuste

;Si el valor de las centesimas es 0, el decremento coloca el registro en 99

;Si el valor de los segundos o minutos es 0, el decremento coloca el registro en 59

```

DEC_FILE    movlw centesimas

            xorwf cursor, W

            btfsc STATUS, Z

            goto DEC_LIM99

            goto DEC_LIM59

DEC_LIM99 decf INDF1, F

            movf INDF1, W

            xorlw .255

            btfss STATUS, Z

            return

            movlw .99

            movwf INDF1

            return

DEC_LIM59 decf INDF1, F

            movf INDF1, W

            xorlw .255

            btfss STATUS, Z

            return

            movlw .59

            movwf INDF1

            return

```

;Rota el cursor de ajuste a la izquierda

```
RL_CURSOR    movlw minutos

              xorwf cursor, W

              btfsc STATUS, Z

              goto CHECK_RLCUR

              incf cursor, F

              return
```

;Comprueba si los registros ya tienen datos

;sino el cursor vuelve al display de centesimas

```
CHECK_RLCUR  movlw .0

              xorwf centesimas, W

              btfss STATUS, Z

              bsf          banderas, 3

              movlw .0

              xorwf segundos, W

              btfss STATUS, Z

              bsf          banderas, 3

              movlw .0

              xorwf minutos, W

              btfss STATUS, Z

              bsf          banderas, 3

              btfss banderas, 3
```

```

goto    $+6

movf    minutos, W

xorwf   cursor, W

btfss   STATUS, Z

goto    $+2

goto    TEMPORIZADOR

movlw   0x20

movwf   cursor

return

```

```

RR_CURSOR    decf    cursor, F

              banksel    IOCCF

              return

```

;Mostrando los dígitos en los display (bcd\_decena, bcd\_unidad)

```

SHOW_DISP    movf    INDF0, W

              call    DECIMAL_A_BCD

              btfsc   banderas, 0

              goto    SHOW_UNI

              goto    SHOW_DEC

```

```

SHOW_UNI     movf    bcd_unidad, W

              call    TABLA_7SEG_AC

```

```

    btfsc banderas, 4
    call   BCD_PARP
    movwf PORTB
    call   PUNTOS
    bcf     banderas, 0
    return

```

```

SHOW_DEC movf bcd_decena, W
    call   TABLA_7SEG_AC
    btfsc banderas, 4
    call   BCD_PARP
    movwf PORTB
    call   PUNTOS
    bsf     banderas, 0
    return

```

;Activa el display a la izq y lo deja así  $\frac{1}{2}$  por 1ms (base de tiempo del TMR0)

;barrido = PORTA

```

ROTA_DISP rlf     barrido, F
    btfsc barrido, 6
    goto  CORREGIR
    btfsc banderas, 0
    incf   current_display, F

```

```
    movf  current_display, W
    movwf FSR0L
    return
```

CORREGIR movlw 0x01

```
    movwfbarrido
    movlw      0x20
    movwfcurent_display
    movwfFSR0L
    return
```

CRONOMETRO    btfss banderas, 2

```
                goto  RESET_TMR
                decfsz cont_ms, F
                goto  RESET_TMR
                call  INCREMENTA
                goto  RESET_TMR
```

INCREMENTA    movlw .10

```
                movwf cont_ms
                movf  centesimas, W
                xorlw .99
```

	btfsc STATUS, Z
	goto INC_SEG
	incf centesimas, F
	return
INC_SEG	clrf centesimas
	movf segundos, W
	xorlw .59
	btfsc STATUS, Z
	goto INC_MIN
	incf segundos, F
	return
INC_MIN	clrf segundos
	movf minutos, W
	xorlw .59
	btfsc STATUS, Z
	goto CLR_INC
	incf minutos, F
	return
CLR_INC	clrf centesimas
	clrf segundos
	clrf minutos
	return



## TEMPORIZADOR

```
btfss banderas, 2  
decfsz cont_ms, F  
goto RESET_TMR  
call DECREMENTA  
goto RESET_TMR
```

```
DECREMENTA bcf banderas, 4  
goto CHECK_DEC
```

```
DECR movlw .10  
  
movwf cont_ms  
movf centesimas, W  
xorlw .0  
btfsc STATUS, Z  
goto DEC_SEG  
decf centesimas, F  
return
```

```
DEC_SEG movlw .99  
movwf centesimas  
movf segundos, W  
xorlw .0
```

```
goto DEC_MIN
decf segundos, F
return
```

```
goto DEC_MIN
```

```
defc segundos, F
```

return

DEC\_MIN movlw .59

movwf segundos

```
decf minutos, F
```

Se verifica si todos los registros están en cero (minutos=0, segundos=0, centesimas=0)

```
CHECK_DEC    movf  minutos, W
```

```
xorlw .0
```

btfss STATUS, Z

goto DECR

```
movf segundos, W
```

```
xorlw .0
```

btfss STATUS, Z

goto DECRC

movf centesimas, W

```
xorlw .0
```

btfss STATUS, Z

goto DECR

bcf	banderas, 3
-----	-------------

```

                                bsf          banderas, 4

                                movlw centesimas

                                movwfcursor

                                bsf          banderas, 6 ;Si todos los registro llegan a cero se activa
la alarma

                                retfie

```

;Se hacen varias temporizaciones "anidadadas", cada dos temporizaciones de 100ms

;se alterna el bit 4 del puerto C, haciendo esto 50 veces la alarma dura 10s

```

ALARMA          decfsz alarma3_ms, F

                goto  RESET_TMR

                movlw .100

                movwf alarma3_ms

                decfsz alarma2_ms, F

                goto  ALARMA

                call  TOGGLE_BEEP

                movlw .2

                movwf alarma2_ms

                decfsz alarma1_ms, F

                goto  ALARMA

                bcf          banderas, 6

                movlw .50

```

```

movwf alarma1_ms
bcf      PORTC, 4

goto    RESET_TMR

TOGGLE_BEEP btfss PORTC, 4

goto    $+3

bcf      PORTC, 4

return

bsf      PORTC, 4

return

```

;Luego de la interrupci3n por cambio de nivel, si est3 transcurriendo el tiempo de alarma

;se detiene colocando a cero el bit 4 del puerto C

CHECK\_ALARMA

```

btfss    banderas, 6

return

bcf      banderas, 6

bcf      PORTC, 4

goto    RESET_TMR

```

;Si est3 en modo ajuste, env3a al display donde se encuentra el cursor

;0xff por 60ms y luego el dato que deber  ir al display por otros 60ms

BCD\_PARP movwfparp\_display

movf FSR0L, W

xorwf cursor, W

btfsc STATUS, Z

goto COMP\_PARP

movf parp\_display, W

return

COMP\_PARP decfszparp\_ms, F

goto FIN\_PARP

movlw .60

movwfparp\_ms

btfsc banderas, 5

goto PARP\_ON

goto PARP\_OFF

PARP\_ON bcf banderas, 5

movf parp\_display, W

return

PARP\_OFF bsf banderas, 5

movlw0xff

return

FIN\_PARP btfss banderas, 5

goto \$+3

```
    movlw 0xff
    return

    movf   parp_display, W
    return
```

;Separa los digitos de un registro

;deja la decena en bcd\_decena y la unidad en bcd\_unidad

DECIMAL\_A\_BCD

```
    movwfbcd_unidad
    clrf   bcd_decena
```

RESTA\_DECENA

```
    movlw .10
    subwfbcd_unidad
    btfss  STATUS, C
    goto   SUMA_UNIDAD
    incf   bcd_decena, F
    goto   RESTA_DECENA
```

```
SUMA_UNIDAD  addwfbcd_unidad, F
              return
```

;Muestra si estÃ¡ en modo CronÃ³metro o Temporizador a travÃ©s del bit 5 del puerto C

```

SHOW_STATUS btfss    banderas, 1

                goto  IS_TEMP_OFF

                bsf      PORTC, 5

                return

IS_TEMP_OFF    bcf      PORTC, 5

                return

```

```

PUNTOS          btfss  banderas, 0

                return

                movf  current_display, W

                xorlw centesimas

                btfsc STATUS, Z

                return

                bcf      PORTB, 7

                return

```

;Tabla display 7seg i<sub>0</sub> 1/2nodo comi<sub>0</sub> 1/2n

```

TABLA_7SEG_AC

                brw

                retlw b'11000000' ; CERO

                retlw b'11111001' ; UNO

```

```
retlw b'10100100' ; DOS
retlw b'10110000' ; TRES
retlw b'10011001' ; CUATRO
retlw b'10010010' ; CINCO
retlw b'10000010' ; SEIS
retlw b'11111000' ; SIETE
retlw b'10000000' ; OCHO
retlw b'10010000' ; NUEVE
```

```
end
```