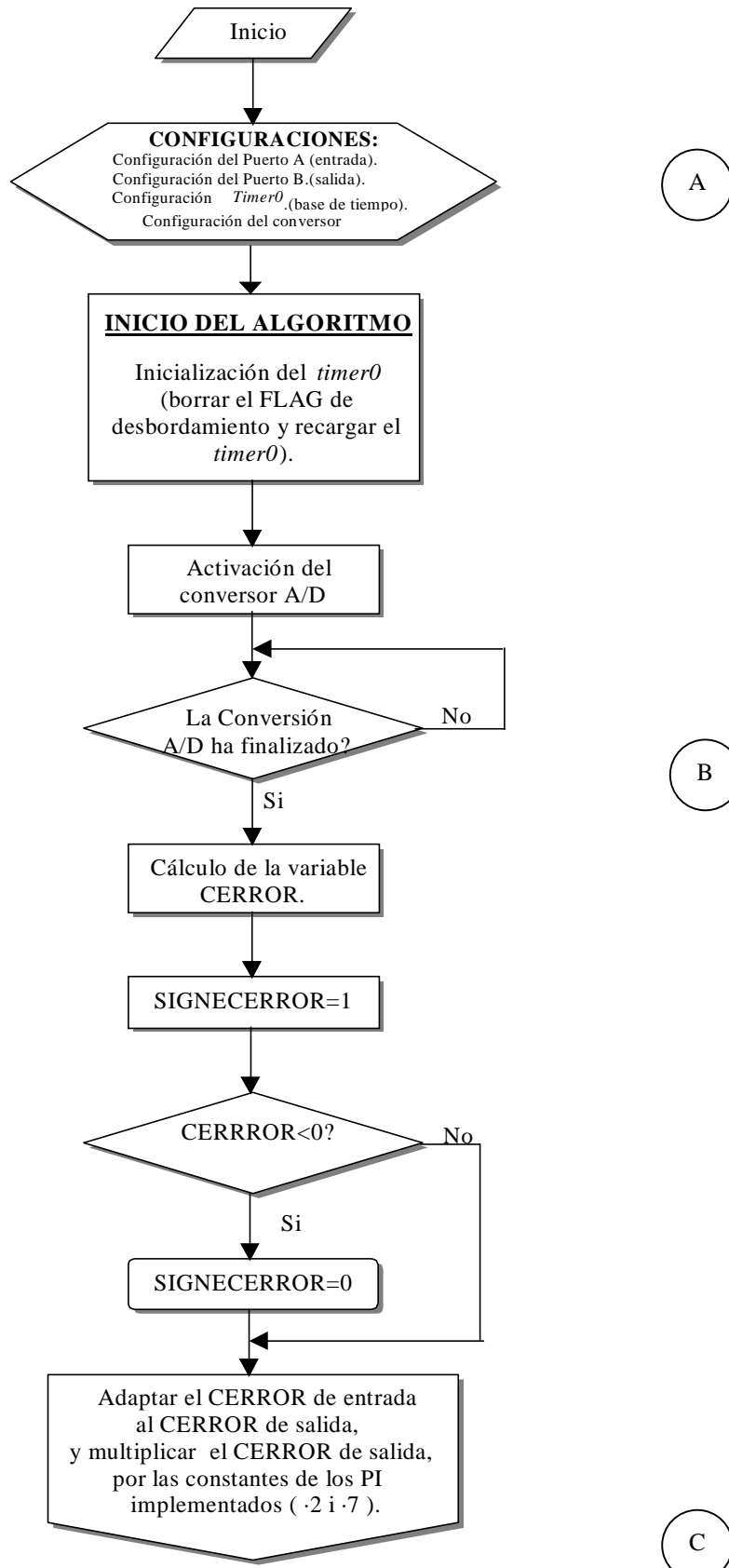
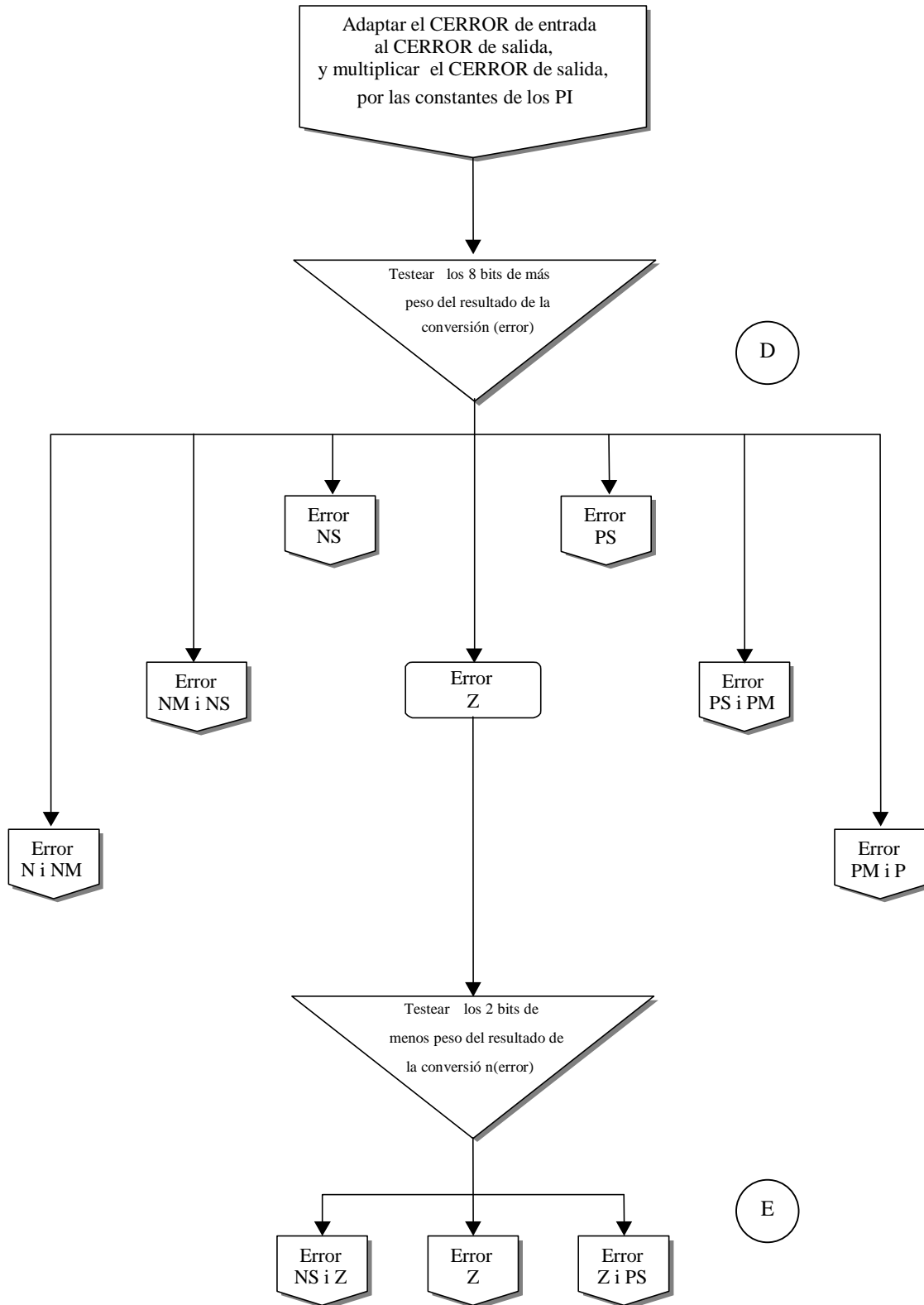
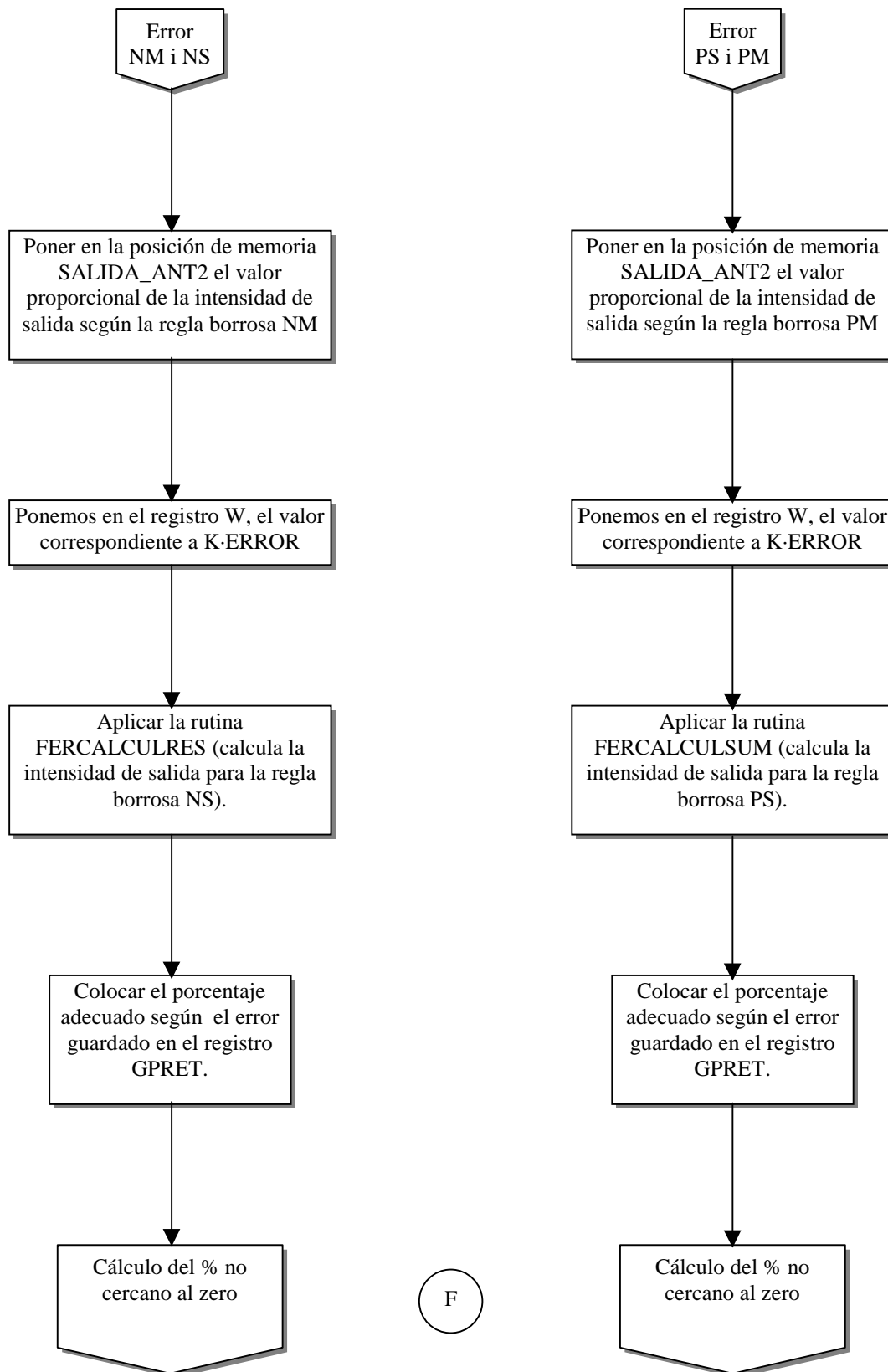


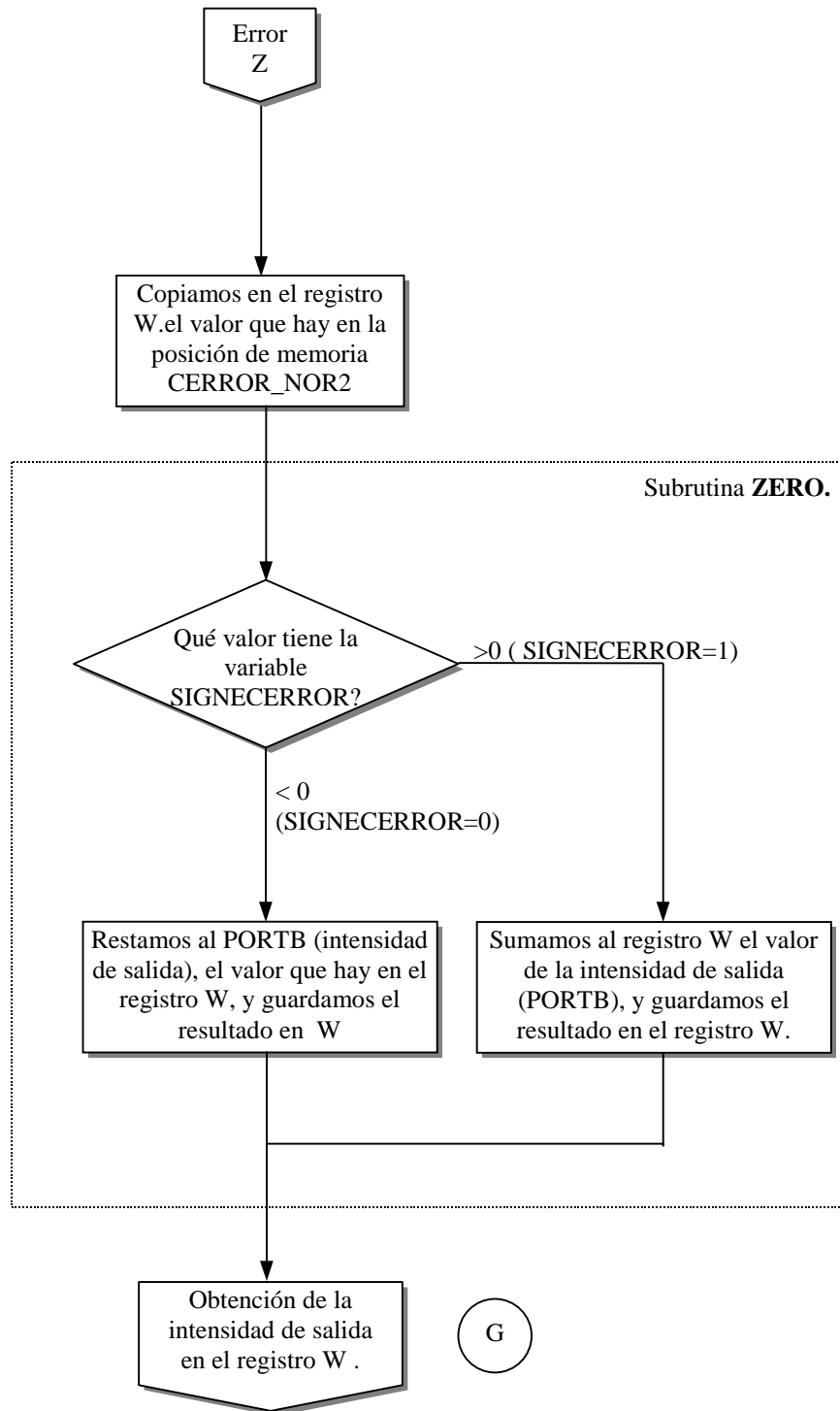
Anexo

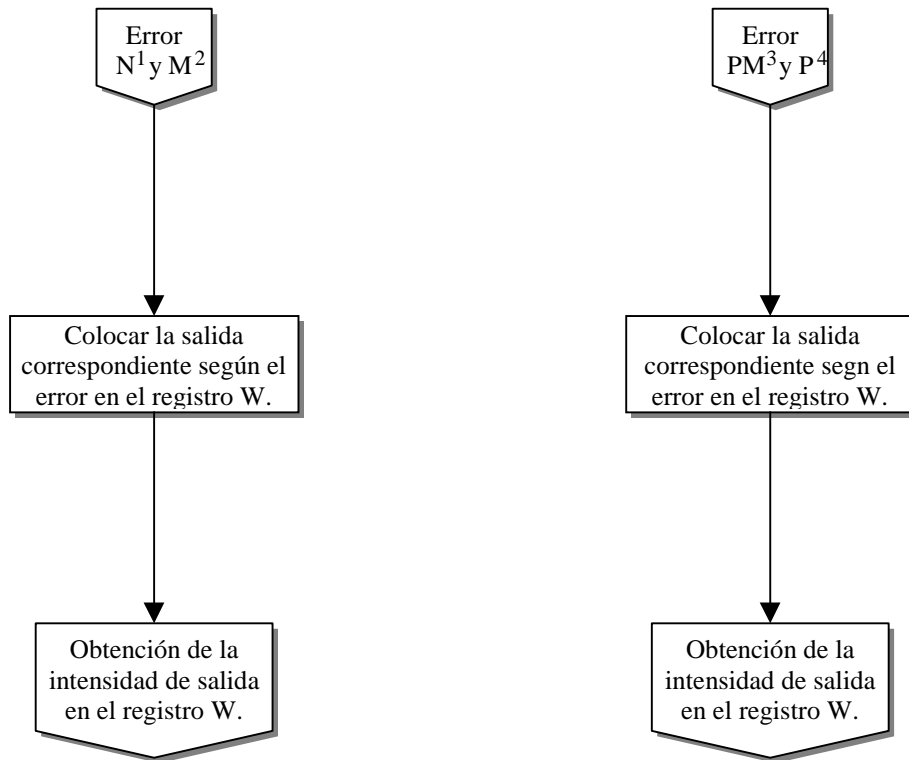
A.1 Diagrama de flujo del controlador borroso









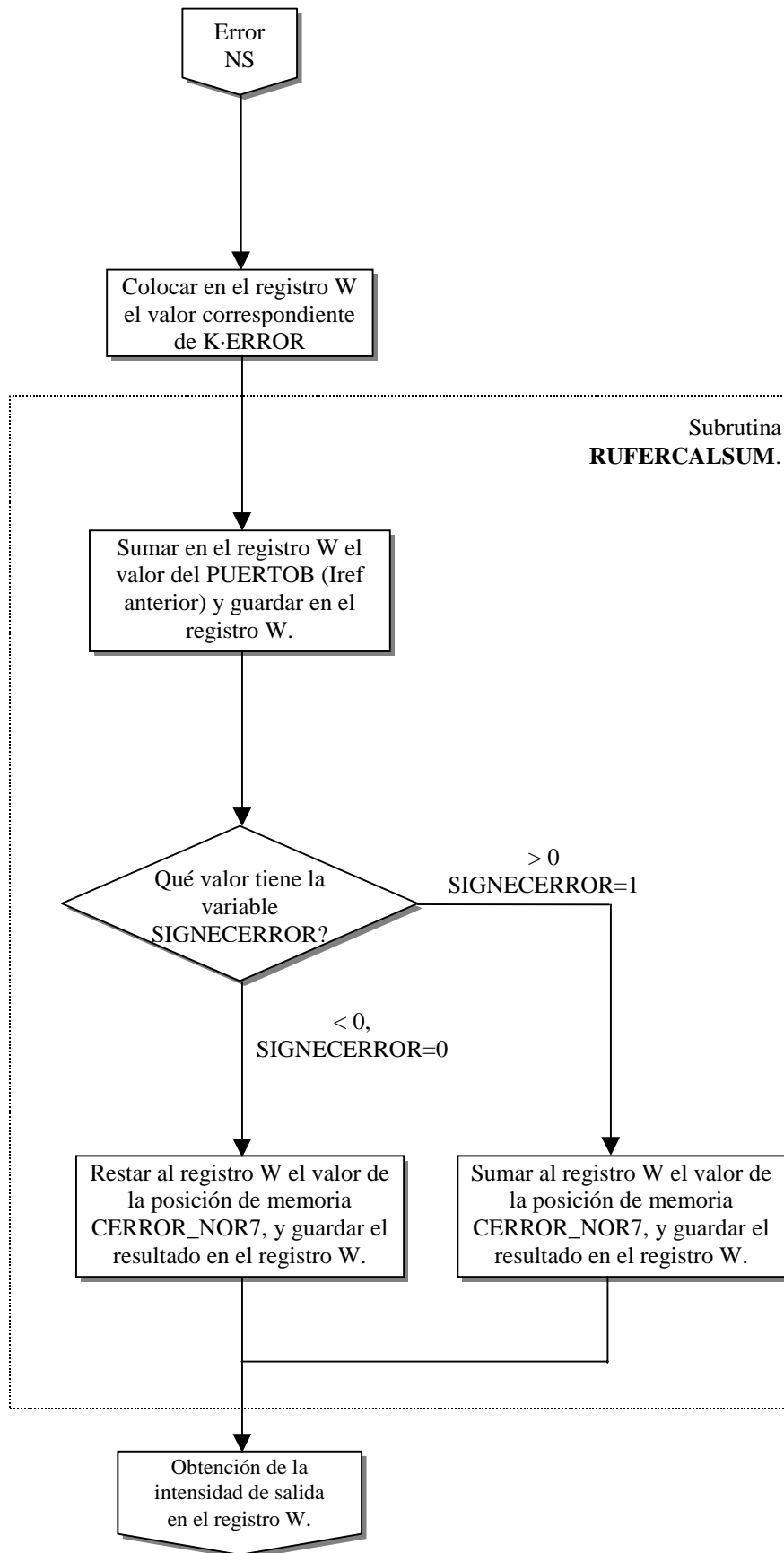


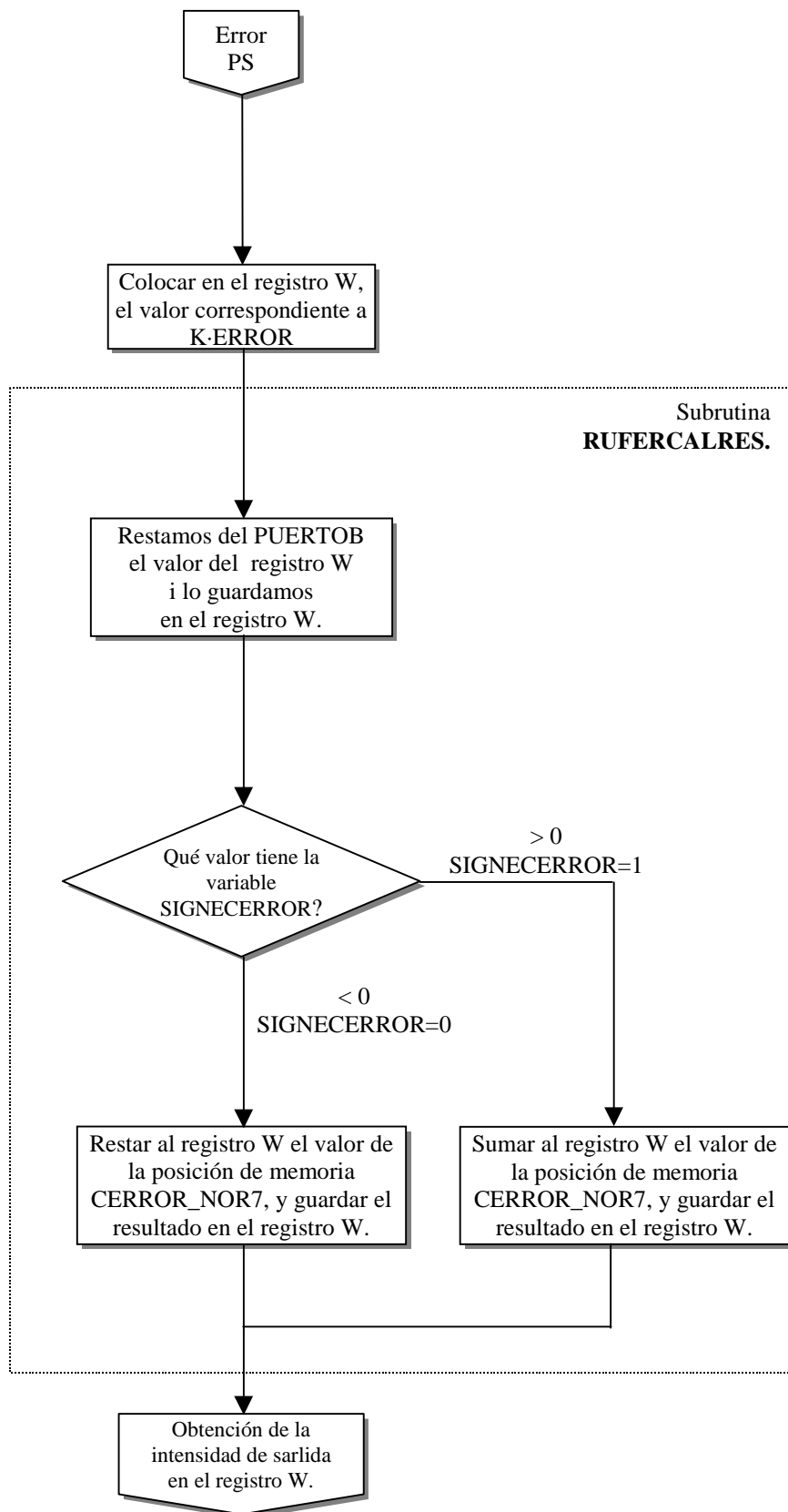
¹ El conjunto borroso N posee como consecuente el valor $I_{ref}(k)=0.4$ A.

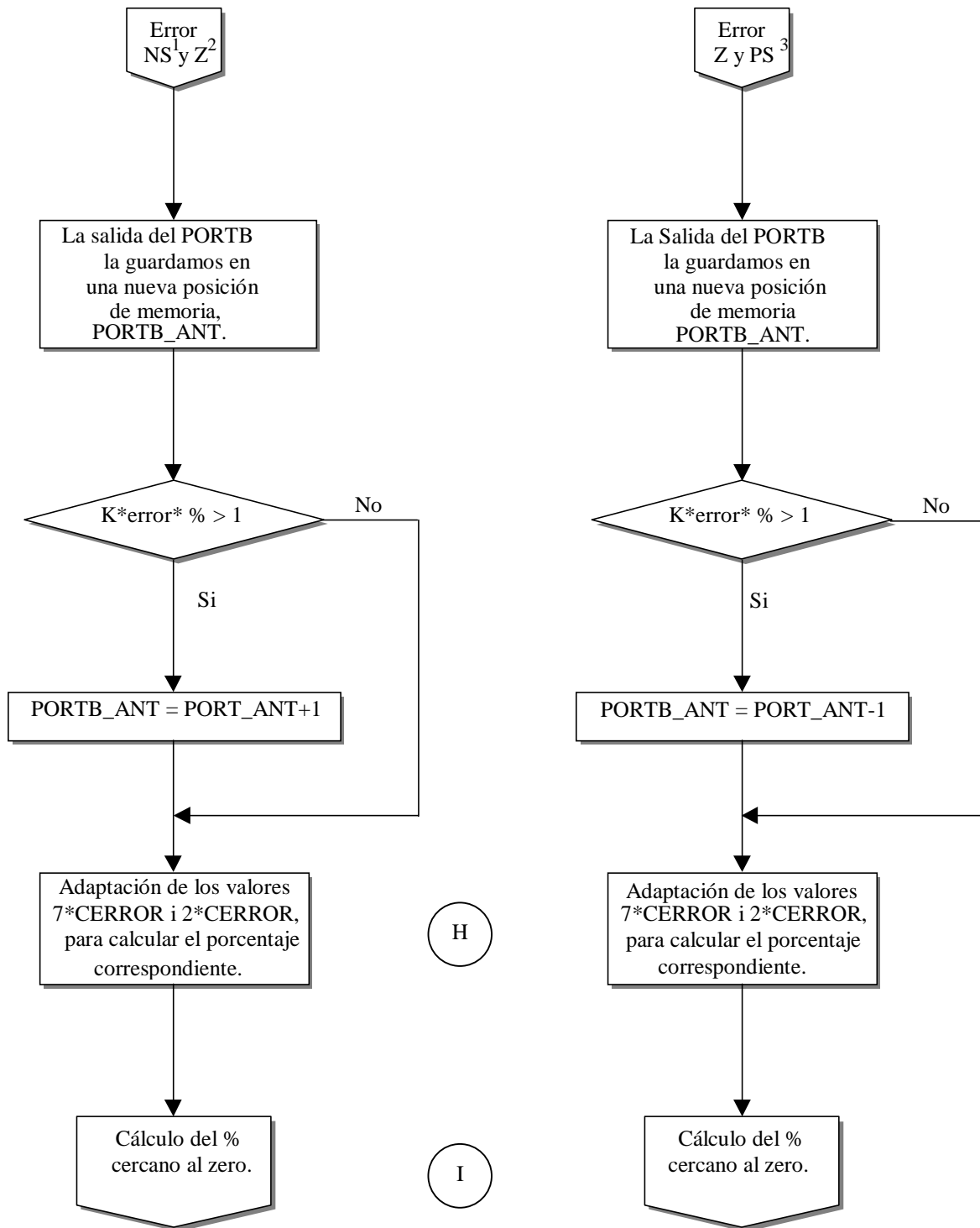
² El conjunto borroso NM posee como consecuente el valor $I_{ref}(k)=1$ A.

³ El conjunto borroso PM posee como consecuente el valor $I_{ref}(k)=1.6$ A.

⁴ El conjunto borroso P posee como consecuente el valor $I_{ref}(k)=2.4$ A.



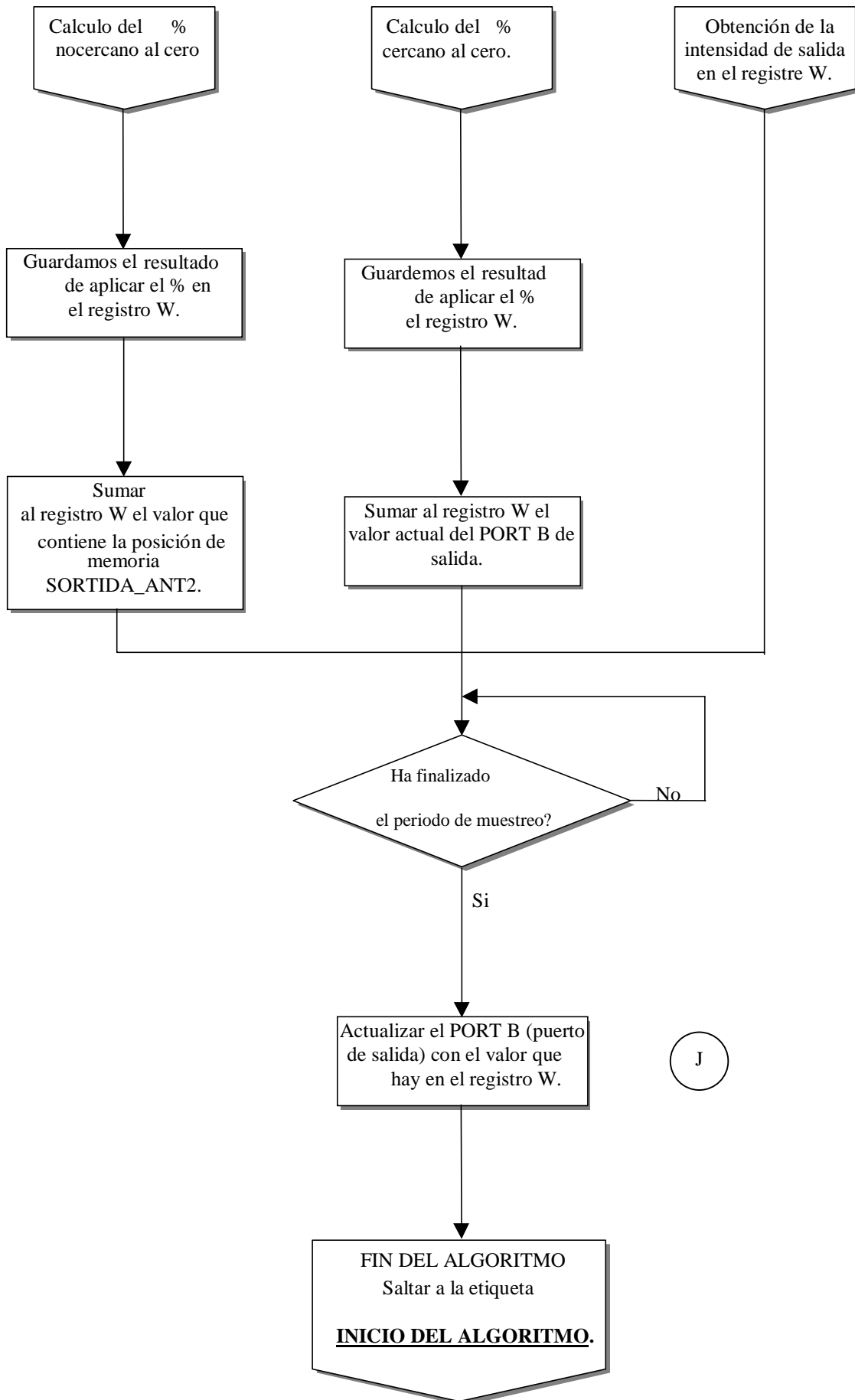




¹ El conjunto borroso NS tiene consecuente la ecuación $Iref(k) = 0.72 \cdot error + 7 \cdot CERROR + Iref(k-1)$

² El conjunto borroso Z tiene consecuente la ecuación $Iref(k) = 2 \cdot CERROR + Iref(k-1)$

³ El conjunto borroso PS tiene consecuente la ecuación $Iref(k) = 0.72 \cdot error + 7 \cdot CERROR + Iref(k-1)$



A.2 Código del programa

```

;*****
;***** Directivas del programa *****
;*****

LIST P=16F873
INCLUDE "P16F873.INC"

;*****
;***** Etiquetas del programa *****
;*****

SIGNECERROR EQU 0x20 ;Registro de 8 bits donde sólo utilizamos
;el bit de menos peso, y este nos indica
;el signo de la variable CERROR. Si el
;bit de menos peso (bit 0) es 1, el signo
;de la variable CERROR es negativo. Si es
;0 el signo de la variable es positivo.

ERROR_ANTERIOR EQU 0x21 ;Posición de memoria donde guardamos el
;valor anterior del error, para poder
;calcular posteriormente el valor de la
;variable CERROR (cambio en el error).
;CERROR= ERROR actual-ERROR anterior.

CERROR EQU 0x29 ;Registro donde queda almacenado el valor
;de la variable (CERROR).

CERROR_NOR2 EQU 0x22
CERROR_NOR7 EQU 0x23 ;Posiciones de memoria donde se
;almacenará el valor de CERROR
;multiplicado por las constantes
;respectivas (*2 i *7) de sus PI.

OUT1 EQU 0x25 ;Posición de memoria donde se guarda el
;resultado de una función de pertenencia.

OUT2 EQU 0x26 ;Posición de memoria donde se guarda el
;resultado de la otra función de
;pertenencia.

GRPERT EQU 0x24 ;Registro de 8 bits, donde se guarda el
;Grado de PERTenencia a aplicar a los
;registros OUT1 y OUT2.

SALIDA_ANT1 EQU 0x27 ;Posición de memoria donde está
;almacenado el valor de una funció de
;pertenencia multiplicat por su grado de
;pertenencia.

SALIDA_ANT2 EQU 0x28 ;Posición de memoria donde está
;almacenado el otro valor de la función
;de pertenencia multiplicado por su grado
;de pertinencia.

PORTB_ANT EQU 0x2A ;Registro donde se guarda el valor de
;anterior de salida(Intensidad de
;referencia anterior)

```

```

AUX          EQU    0x2B ;Posición de memoria auxiliar utilizada
                    ;para los calculos en el resto de la
                    ;variable CERROR.

                ORG    0          ;El programa empieza en la dirección 0
                    ;(Vector de reset).
                GOTO  inici      ;El programa salta la etiqueta inici, que
                ORG    5          ;se encuentra en la dirección 5,de esta
                    ;manera saltamos el vector de
                    ;interrupciones.

;*****
;***** Inicializaciones *****
;*****
;EN EL DIAGRAMA DE FLUJO CORRESPONDE AL ESTADO A

inici:         bsf    STATUS,RP0 ;selección del banco 1.

                movlw b'00001110' ;Configuración del registro "ADCON1",
                movwf ADCON1      ;el bit7 → El resultado de la
                    ;conversión A/D esta justificado a la
                    ;izquierda
                    ;ADRESH:ADRESL (XXXX XXXX:XX00 0000).
                    ;los bits <6:4> →no están
                    ;implementados.
                    ;los bits <3:0> → entrada analógica
                    ;(RA0) y4 bits de entradas/salidas
                    ;digitales.

                movlw b'10001000' ;Configuración del registro "OPTION_REG"
                movwf OPTION_REG ;el bit 7→deshabilita los transistores
                    ;de pull-ups del PORTB.
                    ;El bit 6→no se utiliza.
                    ;El bit 5→El reloj del timer0 es el
                    ;mismo que el reloj del microcontrolador,
                    ;no requiere reloj externo.
                    ;El bit 4→no se utiliza.
                    ;los bits <3:0>→ Configuran el timer0
                    ;para que en cada ciclo de maquina se
                    ;incremente el registro que controla el
                    ;tiempo.

                movlw 0x01        ;Configuramos el portA como 1 entrada y
                movwf TRISA       ;5 sortides, seleccionamos el canal 0
                    ;(RA0) como entrada y el resto como
                    ;salidas digitales.

                clrfs TRISB       ;Configuramos el portB como salidas.

                bcf    STATUS,RP0 ;selección del banco 0.

                movlw b'10000001' ;Configuración del registro "ADCON0",
                movwf ADCON0      ;los bits <7:6> → la frecuencia de
                    ;conversión
                    ;(A/D)es Fosc/32 ( Fosc = reloj del
                    ;sistema, 20 MHz)per bit.
                    ;Los bits <5:3> → selecciona el canal de
                    ;la señal analógica a convertir canal0
                    ;(RA0).
                    ;el bit 2 → indica el inicio y el final

```

```

;de la conversión A/D.
;el bit 1 → no está implementado.
;el bit 0 → indica si el proceso de
;conversión (A/D) está operativo o no.

movlw 0xFF          ;Inicializamos el portB(puerto de salida)
movwf PORTB        ;con la intensidad máxima (0xFF →2,4 A).
;*****
;***** Programa Principal *****
;*****

principi:  clrf  INTCON      ;borramos el flag del timer0, que estaba
; a 1 y nos indicaba el desbordamiento
;del timer0.

movlw 0x09         ;carga del registro "TMR0" con el valor
movwf TMR0        ;9 para que se desborde cada 50 µseg.

conversio:  bsf   ADCON0,GO  ;Inicio de la conversión A/D, GO=1.
            btfsc ADCON0,GO  ;Mientras el bit GO sea igual a 1 el
            goto  conversio  ; programa estará en el bucle, cuando
; valgui 0, saldrá del bucle, lo que
; indicará que el valor digital que hay en
; los registros ADRESH i ADRESL es válido.

;***** cálculo_cerror *****
;EN EL DIAGRAMA DE FLUJO CORRESPONDE AL ESTADO B

            clrf  SIGNECERROR
            incf  SIGNECERROR,F      ;Inicializamos el signo del
; CERROR=1 que implica CERROR
;positivo.

            movf  ERROR_ANTERIOR,W  ;Calculamos el
            subwf ADRESH,W          ;CERROR = ERRORactual - ERRORanterior.
            btfsc STATUS,C          ;Si el resultado del CERROR es
            goto  ficerror          ;positivo saltamos a la etiqueta
;ficerror,

            xorlw 0xFF              ;Si es negativo, hacemos el
            addlw 0x01              ;complemento a 2 para obtener el
            clrf  SIGNECERROR      ;módulo correcto y borramos la
; variable SIGNECERROR, que implica
;CERROR negativo.

ficerror:   movwf CERROR            ;Guardamos la variable CERROR que
; está en el registro de trabajo W
; en la posición de memoria
; correspondiente.

;Realizamos un testeo de la variable CERROR, y según su valor le
;asignamos el valor de salida correspondiente.
;La variable CERROR está tabulada a un valor máximo de 64 estados.
;pero como en la variable CERROR despreciamos los dos bits de menor
;peso, el número de estados posibles se reduce a 16.

            btfsc CERROR,4         ;si el bit 4 es 1 saltamos a la etiqueta
            goto  c16              ;c16.
            btfss CERROR,3         ;si el valor es 0 testeamos el bit3, si
            goto  de0a7            ;este es 0 vamos a la etiqueta de0a7.
            btfss CERROR,2         ;si bit3 es 1, miramos el bit2
            goto  de8a11           ;si bit2 es 0 vamos a la etiqueta de8a11
            btfss CERROR,1         ;si valor bit2 es 1, miramos el bit1
            goto  de12a13         ;si bit1 es 0 vamos a la etiqueta de12a13

```

```

                btfss CERROR,0    ;si bit1 es 1, miramos el bit0
                goto  c14         ;si bit0 es 0 saltamos a la etiqueta C14
                goto  c15         ;si bit0 es 1 saltamos a la etiqueta C15

de0a7:         btfss CERROR,2    ;....
                goto  de0a3       ;...
                btfss CERROR,1    ;..
                goto  de4a5
                btfss CERROR,0
                goto  c6
                goto  c7
de0a3:         btfss CERROR,1    ;....
                goto  de0a1
                btfss CERROR,0
                goto  c2
                goto  c3
de8a11:        btfss CERROR,1
                goto  de8a9
                btfss CERROR,0
                goto  c10
                goto  c11
de4a5:         btfss CERROR,0
                goto  c4
                goto  c5
de0a1:         btfss CERROR,0
                goto  c0
                goto  c1
de12a13:       btfss CERROR,0
                goto  c12
                goto  c13
de8a9:         btfss CERROR,0
                goto  c8
                goto  c9

```

;Una vez encontrado el valor del CERROR, lo adaptamos según las ecuaciones del consecuente. El CERROR tiene un valor mínimo de 0 y un valor máximo de 0.05. Esto implica valores máximos y mínimos en los consecuentes de 0 a 35 y de 0 a 10.

;EN EL DIAGRAMA DE FLUJO CORRESPONDE AL ESTADO (c)

```

c0:           movlw d'0'
                movwf CERROR_NOR2
                movlw d'0'
                movwf CERROR_NOR7
                goto  nova
c1:           movlw d'1'
                movwf CERROR_NOR2
                movlw d'3'
                movwf CERROR_NOR7
                goto  nova
c2:           movlw d'2'
                movwf CERROR_NOR2
                movlw d'5'
                movwf CERROR_NOR7
                goto  nova
c3:           movlw d'2'
                movwf CERROR_NOR2
                movlw d'7'
                movwf CERROR_NOR7
                goto  nova
c4:           movlw d'3'

```

```
        movwf CERROR_NOR2
        movlw d'9'
        movwf CERROR_NOR7
        goto nova
c5:     movlw d'3'
        movwf CERROR_NOR2
        movlw d'11'
        movwf CERROR_NOR7
        goto nova
c6:     movlw d'4'
        movwf CERROR_NOR2
        movlw d'13'
        movwf CERROR_NOR7
        goto nova
c7:     movlw d'4'
        movwf CERROR_NOR2
        movlw d'15'
        movwf CERROR_NOR7
        goto nova
c8:     movlw d'5'
        movwf CERROR_NOR2
        movlw d'18'
        movwf CERROR_NOR7
        goto nova
c9:     movlw d'6'
        movwf CERROR_NOR2
        movlw d'20'
        movwf CERROR_NOR7
        goto nova
c10:    movlw d'7'
        movwf CERROR_NOR2
        movlw d'22'
        movwf CERROR_NOR7
        goto nova
c11:    movlw d'7'
        movwf CERROR_NOR2
        movlw d'24'
        movwf CERROR_NOR7
        goto nova
c12:    movlw d'8'
        movwf CERROR_NOR2
        movlw d'26'
        movwf CERROR_NOR7
        goto nova
c13:    movlw d'8'
        movwf CERROR_NOR2
        movlw d'28'
        movwf CERROR_NOR7
        goto nova
c14:    movlw d'9'
        movwf CERROR_NOR2
        movlw d'30'
        movwf CERROR_NOR7
        goto nova
c15:    movlw d'9'
        movwf CERROR_NOR2
        movlw d'32'
        movwf CERROR_NOR7
        goto nova
c16:    movlw d'10'
        movwf CERROR_NOR2
```

```

        movlw d'35'
        movwf CERROR_NOR7
        goto nova
nova:    movf  ADRESH,W           ;Guardamos el valor del error
        movwf ERROR_ANTERIOR ; actual en la variable
                                           ; error_anterior, para calcular el
                                           ;próximo CERROR

;*****
;***** algoritmo de búsqueda de error *****
;*****

;EN EL DIAGRAMA DE FLUJO CORRESPONDE AL ESTADO D
;Una vez tengamos el error en los registros de salida ADRESH y ADRESL,
;con la configuración justificada a la izquierda, es decir,XXXX XXXX :
;XX00 ;0000, testeamos los 8 bits de más peso, para obtener el error y
borrosificarlo.
;***** Secuencia 1 *****

busqueda:  btfss ADRESH,7
           goto menor0
           btfss ADRESH,6
           goto e200i2FF

           btfss ADRESH,5
           goto e300i37F
           btfss ADRESH,4
           goto e380i3BF
           btfss ADRESH,3
           goto e3C0i3DF
           btfss ADRESH,2
           goto e3E0i3EF
           btfss ADRESH,1
           goto e3F0i3F7
           btfss ADRESH,0
           goto e3F8i3FB
           goto e3FCi3FF
;***** Secuencia 1 *****

menor0:   btfss ADRESH,6
           goto e000i0FF

           btfss ADRESH,5
           goto e100i17F
           btfss ADRESH,4
           goto e180i1BF
           btfss ADRESH,3
           goto e1C0i1DF
           btfss ADRESH,2
           goto e1E0i1EF
           btfss ADRESH,1
           goto e1F0i1F7
           btfss ADRESH,0
           goto e1F8i1FB
           goto e1FCi1FF
;***** Secuencia 1 *****

e200i2FF: btfss ADRESH,5
           goto e200i27F
           btfss ADRESH,4
           goto e280i2BF
           btfss ADRESH,3

```

```

        goto e2C0i2DF
        btfss ADRESH,2
        goto e2E0i2EF
        btfss ADRESH,1
        goto e2F0i2F7
        btfss ADRESH,0
        goto e2F8i2FB
        goto e2FCi2FF
;***** Secuencia 1 *****
e000i0FF:  btfss ADRESH,5
           goto e000i07F
           btfss ADRESH,4
           goto e080i0BF
           btfss ADRESH,3
           goto e0C0i0DF
           btfss ADRESH,2
           goto e0E0i0EF
           btfss ADRESH,1
           goto e0F0i0F7
           btfss ADRESH,0
           goto e0F8i0FB
           goto e0FCi0FF
;*****4 ELEMENTOS*****
;***** Secuencia 1 *****
e300i37F:  btfss ADRESH,4
           goto e300i33F
           btfss ADRESH,3
           goto e340i35F
           btfss ADRESH,2
           goto e360i36F
           btfss ADRESH,1
           goto e370i377
           btfss ADRESH,0
           goto e378i37B
           goto e37Ci37F
;***** Secuencia 2 *****
e100i17F:  btfss ADRESH,4
           goto e100i13F
           btfss ADRESH,3
           goto e140i15F
           btfss ADRESH,2
           goto e160i16F
           btfss ADRESH,1
           goto e170i177
           btfss ADRESH,0
           goto e178i17B
           goto e17Ci17F
;***** Secuencia 3 *****
e200i27F:  btfss ADRESH,4
           goto e200i23F
           btfss ADRESH,3
           goto e240i25F
           btfss ADRESH,2
           goto e260i26F
           btfss ADRESH,1
           goto e270i277
           btfss ADRESH,0
           goto e278i27B
           goto e27Ci27F
;***** Secuencia 4 *****
e000i07F:  btfss ADRESH,4

```



```
        goto e000i03F
        btfss ADRESH,3
        goto e040i05F
        btfss ADRESH,2
        goto e060i06F
        btfss ADRESH,1
        goto e070i077
        btfss ADRESH,0
        goto e078i07B
        goto e07Ci07F

;*****FINAL DE LOS 4 ELEMENTOS*****
;*****INICIO DE LOS 8 ELEMENTOS*****
;***** Secuencia 1 *****
e380i3BF:  btfss ADRESH,3
          goto e380i39F
          btfss ADRESH,2
          goto e3A0i3AF
          btfss ADRESH,1
          goto e3B0i3B7
          btfss ADRESH,0
          goto e3B8i3BB
          goto e3BCi3BF
;***** Secuencia 2 *****
e180i1BF:  btfss ADRESH,3
          goto e180i19F
          btfss ADRESH,2
          goto e1A0i1AF
          btfss ADRESH,1
          goto e1B0i1B7
          btfss ADRESH,0
          goto e1B8i1BB
          goto e1BCi1BF
;***** Secuencia 3 *****
e280i2BF:  btfss ADRESH,3
          goto e280i29F
          btfss ADRESH,2
          goto e2A0i2AF
          btfss ADRESH,1
          goto e2B0i2B7
          btfss ADRESH,0
          goto e2B8i2BB
          goto e2BCi2BF
;***** Secuencia 4 *****
e080i0BF:  btfss ADRESH,3
          goto e080i09F
          btfss ADRESH,2
          goto e0A0i0AF
          btfss ADRESH,1
          goto e0B0i0B7
          btfss ADRESH,0
          goto e0B8i0BB
          goto e0BCi0BF
;***** Secuencia 5 *****
e300i33F:  btfss ADRESH,3
          goto e300i31F
          btfss ADRESH,2
          goto e320i32F
          btfss ADRESH,1
          goto e330i337
          btfss ADRESH,0
```

```
        goto e338i33B
        goto e33Ci33F
;***** Secuencia 6 *****
e100i13F:  btfss ADRESH,3
           goto e100i11F
           btfss ADRESH,2
           goto e120i12F
           btfss ADRESH,1
           goto e130i137
           btfss ADRESH,0
           goto e138i13B
           goto e13Ci13F
;***** Secuencia 7 *****
e200i23F:  btfss ADRESH,3
           goto e200i21F
           btfss ADRESH,2
           goto e220i22F
           btfss ADRESH,1
           goto e230i237
           btfss ADRESH,0
           goto e238i23B
           goto e23Ci23F
;***** Secuencia 8 *****
e000i03F:  btfss ADRESH,3
           goto e000i01F
           btfss ADRESH,2
           goto e020i02F
           btfss ADRESH,1
           goto e030i037
           btfss ADRESH,0
           goto e038i03B
           goto e03Ci03F
;***** INICIO DE LOS 16 ELEMENTOS *****
;***** Secuencia 1 *****
e3C0i3DF:  btfss ADRESH,2
           goto e3C0i3CF
           btfss ADRESH,1
           goto e3D0i3D7
           btfss ADRESH,0
           goto e3D8i3DB
           goto e3DCi3DF
;***** Secuencia 2 *****
e1C0i1DF:  btfss ADRESH,2
           goto e1C0i1CF
           btfss ADRESH,1
           goto e1D0i1D7
           btfss ADRESH,0
           goto e1D8i1DB
           goto e1DCi1DF
;***** Secuencia 3 *****
e2C0i2DF:  btfss ADRESH,2
           goto e2C0i2CF
           btfss ADRESH,1
           goto e2D0i2D7
           btfss ADRESH,0
           goto e2D8i2DB
           goto e2DCi2DF
;***** Secuencia 4 *****
e0C0i0DF:  btfss ADRESH,2
           goto e0C0i0CF
           btfss ADRESH,1
```

```
        goto e0D0i0D7
        btfss ADRESH,0
        goto e0D8i0DB
        goto e0DCi0DF
;***** Secuencia 5 *****
e340i35F:  btfss ADRESH,2
           goto e340i34F
           btfss ADRESH,1
           goto e350i357
           btfss ADRESH,0
           goto e358i35B
           goto e35Ci35F
;***** Secuencia 6 *****
e140i15F:  btfss ADRESH,2
           goto e140i14F
           btfss ADRESH,1
           goto e150i157
           btfss ADRESH,0
           goto e158i15B
           goto e15Ci15F
;***** Secuencia 7 *****
e240i25F:  btfss ADRESH,2
           goto e240i24F
           btfss ADRESH,1
           goto e250i257
           btfss ADRESH,0
           goto e258i25B
           goto e25Ci25F
;***** Secuencia 8 *****
e040i05F:  btfss ADRESH,2
           goto e040i04F
           btfss ADRESH,1
           goto e050i057
           btfss ADRESH,0
           goto e058i05B
           goto e05Ci05F
;***** Secuencia 9 *****
e380i39F:  btfss ADRESH,2
           goto e380i38F
           btfss ADRESH,1
           goto e390i397
           btfss ADRESH,0
           goto e398i39B
           goto e39Ci39F
;***** Secuencia 10 *****
e180i19F:  btfss ADRESH,2
           goto e180i18F
           btfss ADRESH,1
           goto e190i197
           btfss ADRESH,0
           goto e198i19B
           goto e19Ci19F
;***** Secuencia 11 *****
e280i29F:  btfss ADRESH,2
           goto e280i28F
           btfss ADRESH,1
           goto e290i297
           btfss ADRESH,0
           goto e298i29B
           goto e29Ci29F
```

```

;***** Secuencia 12 *****
e080i09F:  btfss ADRESH,2
           goto  e080i08F
           btfss ADRESH,1
           goto  e090i097
           btfss ADRESH,0
           goto  e098i09B
           goto  e09Ci09F

;***** Secuencia 13 *****
e300i31F:  btfss ADRESH,2
           goto  e300i30F
           btfss ADRESH,1
           goto  e310i317
           btfss ADRESH,0
           goto  e318i31B
           goto  e31Ci31F

;***** Secuencia 14 *****
e100i11F:  btfss ADRESH,2
           goto  e100i10F
           btfss ADRESH,1
           goto  e110i117
           btfss ADRESH,0
           goto  e118i11B
           goto  e11Ci11F

;***** Secuencia 15 *****
e200i21F:  btfss ADRESH,2
           goto  e200i20F
           btfss ADRESH,1
           goto  e210i217
           btfss ADRESH,0
           goto  e218i21B
           goto  e21Ci21F

;***** Secuencia 16 *****
e000i01F:  btfss ADRESH,2
           goto  e000i00F
           btfss ADRESH,1
           goto  e010i017
           btfss ADRESH,0
           goto  e018i01B
           goto  e01Ci01F

;*****INICIO 32*****
;***** Secuencia 1 *****
e3E0i3EF:  btfss ADRESH,1
           goto  e3E0i3E7
           btfss ADRESH,0
           goto  e3E8i3EB
           goto  e3ECi3EF

;***** Secuencia 2 *****
e1E0i1EF:  btfss ADRESH,1
           goto  e1E0i1E7
           btfss ADRESH,0
           goto  e1E8i1EB
           goto  e1ECi1EF

;***** Secuencia 3 *****
e2E0i2EF:  btfss ADRESH,1
           goto  e2E0i2E7
           btfss ADRESH,0
           goto  e2E8i2EB
           goto  e2ECi2EF

```

```
;***** Secuencia 4 *****
e0E0i0EF:  btfss ADRESH,1
           goto  e0E0i0E7
           btfss ADRESH,0
           goto  e0E8i0EB
           goto  e0ECi0EF
;***** Secuencia 5 *****
e360i36F:  btfss ADRESH,1
           goto  e360i367
           btfss ADRESH,0
           goto  e368i36B
           goto  e36Ci36F
;***** Secuencia 6 *****
e160i16F:  btfss ADRESH,1
           goto  e160i167
           btfss ADRESH,0
           goto  e168i16B
           goto  e16Ci16F
;***** Secuencia 7 *****
e260i26F:  btfss ADRESH,1
           goto  e260i267
           btfss ADRESH,0
           goto  e268i26B
           goto  e26Ci26F
;***** Secuencia 8 *****
e060i06F:  btfss ADRESH,1
           goto  e060i067
           btfss ADRESH,0
           goto  e068i06B
           goto  e06Ci06F
;***** Secuencia 9 *****
e3A0i3AF:  btfss ADRESH,1
           goto  e3A0i3A7
           btfss ADRESH,0
           goto  e3A8i3AB
           goto  e3ACi3AF
;***** Secuencia 10 *****
e1A0i1AF:  btfss ADRESH,1
           goto  e1A0i1A7
           btfss ADRESH,0
           goto  e1A8i1AB
           goto  e1ACi1AF
;***** Secuencia 11 *****
e2A0i2AF:  btfss ADRESH,1
           goto  e2A0i2A7
           btfss ADRESH,0
           goto  e2A8i2AB
           goto  e2ACi2AF
;***** Secuencia 12 *****
e0A0i0AF:  btfss ADRESH,1
           goto  e0A0i0A7
           btfss ADRESH,0
           goto  e0A8i0AB
           goto  e0ACi0AF
;***** Secuencia 13 *****
e320i32F:  btfss ADRESH,1
           goto  e320i327
           btfss ADRESH,0
           goto  e328i32B
           goto  e32Ci32F
```

```

;***** Secuencia 14 *****
e120i12F:  btfss ADRESH,1
           goto  e120i127
           btfss ADRESH,0
           goto  e128i12B
           goto  e12Ci12F
;***** Secuencia 15 *****
e220i22F:  btfss ADRESH,1
           goto  e220i227
           btfss ADRESH,0
           goto  e228i22B
           goto  e22Ci22F
;***** Secuencia 16 *****
e020i02F:  btfss ADRESH,1
           goto  e020i027
           btfss ADRESH,0
           goto  e028i02B
           goto  e02Ci02F
;***** Secuencia 17 *****
e3C0i3CF:  btfss ADRESH,1
           goto  e3C0i3C7
           btfss ADRESH,0
           goto  e3C8i3CB
           goto  e3CCi3CF
;***** Secuencia 18 *****
e1C0i1CF:  btfss ADRESH,1
           goto  e1C0i1C7
           btfss ADRESH,0
           goto  e1C8i1CB
           goto  e1CCi1CF
;***** Secuencia 19 *****
e2C0i2CF:  btfss ADRESH,1
           goto  e2C0i2C7
           btfss ADRESH,0
           goto  e2C8i2CB
           goto  e2CCi2CF
;***** Secuencia 20 *****
e0C0i0CF:  btfss ADRESH,1
           goto  e0C0i0C7
           btfss ADRESH,0
           goto  e0C8i0CB
           goto  e0CCi0CF
;***** Secuencia 21 *****
e340i34F:  btfss ADRESH,1
           goto  e340i347
           btfss ADRESH,0
           goto  e348i34B
           goto  e34Ci34F
;***** Secuencia 22 *****
e140i14F:  btfss ADRESH,1
           goto  e140i147
           btfss ADRESH,0
           goto  e148i14B
           goto  e14Ci14F
;***** Secuencia 23 *****
e240i24F:  btfss ADRESH,1
           goto  e240i247
           btfss ADRESH,0
           goto  e248i24B
           goto  e24Ci24F

```

```

;***** Secuencia 24 *****
e040i04F:  btfss ADRESH,1
           goto  e040i047
           btfss ADRESH,0
           goto  e048i04B
           goto  e04Ci04F
;***** Secuencia 25 *****
e380i38F:  btfss ADRESH,1
           goto  e380i387
           btfss ADRESH,0
           goto  e388i38B
           goto  e38Ci38F
;***** Secuencia 26 *****
e180i18F:  btfss ADRESH,1
           goto  e180i187
           btfss ADRESH,0
           goto  e188i18B
           goto  e18Ci18F
;***** Secuencia 27 *****
e280i28F:  btfss ADRESH,1
           goto  e280i287
           btfss ADRESH,0
           goto  e288i28B
           goto  e28Ci28F
;***** Secuencia 28 *****
e080i08F:  btfss ADRESH,1
           goto  e080i087
           btfss ADRESH,0
           goto  e088i08B
           goto  e08Ci08F
;***** Secuencia 29 *****
e300i30F:  btfss ADRESH,1
           goto  e300i307
           btfss ADRESH,0
           goto  e308i30B
           goto  e30Ci30F
;***** Secuencia 30 *****
e100i10F:  btfss ADRESH,1
           goto  e100i107
           btfss ADRESH,0
           goto  e108i10B
           goto  e10Ci10F
;***** Secuencia 31 *****
e200i20F:  btfss ADRESH,1
           goto  e200i207
           btfss ADRESH,0
           goto  e208i20B
           goto  e20Ci20F
;***** Secuencia 32 *****
e000i00F:  btfss ADRESH,1
           goto  e000i007
           btfss ADRESH,0
           goto  e008i00B
           goto  e00Ci00F
;*****INICIO DE LOS 64 ELEMENTOS*****
;***** Secuencia 1 *****
e3F0i3F7:  btfss ADRESH,0
           goto  e3F0i3F3
           goto  e3F4i3F7
;***** Secuencia 2 *****
e1F0i1F7:  btfss ADRESH,0

```

```

                goto e1F0i1F3
                goto e1F4i1F7
;***** Secuencia 3 *****
e2F0i2F7:      btfss ADRESH,0
                goto e2F0i2F3
                goto e2F4i2F7
;***** Secuencia 4 *****
e0F0i0F7:      btfss ADRESH,0
                goto e0F0i0F3
                goto e0F4i0F7
;***** Secuencia 5 *****
e370i377:      btfss ADRESH,0
                goto e370i373
                goto e374i377
;***** Secuencia 6 *****
e170i177:      btfss ADRESH,0
                goto e170i173
                goto e174i177
;***** Secuencia 7 *****
e270i277:      btfss ADRESH,0
                goto e270i273
                goto e274i277
;***** Secuencia 8 *****
e070i077:      btfss ADRESH,0
                goto e070i073
                goto e074i077
;***** Secuencia 9 *****
e3B0i3B7:      btfss ADRESH,0
                goto e3B0i3B3
                goto e3B4i3B7
;***** Secuencia 10 *****
e1B0i1B7:      btfss ADRESH,0
                goto e1B0i1B3
                goto e1B4i1B7
;***** Secuencia 11 *****
e2B0i2B7:      btfss ADRESH,0
                goto e2B0i2B3
                goto e2B4i2B7
;***** Secuencia 12 *****
e0B0i0B7:      btfss ADRESH,0
                goto e0B0i0B3
                goto e0B4i0B7
;***** Secuencia 13 *****
e330i337:      btfss ADRESH,0
                goto e330i333
                goto e334i337
;***** Secuencia 14 *****
e130i137:      btfss ADRESH,0
                goto e130i133
                goto e134i137
;***** Secuencia 15 *****
e230i237:      btfss ADRESH,0
                goto e230i233
                goto e234i237
;***** Secuencia 16 *****
e030i037:      btfss ADRESH,0
                goto e030i033
                goto e034i037
;***** Secuencia 17 *****
e3D0i3D7:      btfss ADRESH,0
                goto e3D0i3D3

```



```

        goto e3D4i3D7
;***** Secuencia 18 *****
e1D0i1D7:  btfss ADRESH,0
          goto e1D0i1D3
          goto e1D4i1D7
;***** Secuencia 19 *****
e2D0i2D7:  btfss ADRESH,0
          goto e2D0i2D3
          goto e2D4i2D7

;***** Secuencia 20 *****
e0D0i0D7:  btfss ADRESH,0
          goto e0D0i0D3
          goto e0D4i0D7
;***** Secuencia 21 *****
e350i357:  btfss ADRESH,0
          goto e350i353
          goto e354i357
;***** Secuencia 22 *****
e150i157:  btfss ADRESH,0
          goto e150i153
          goto e154i157
;***** Secuencia 23 *****
e250i257:  btfss ADRESH,0
          goto e250i253
          goto e254i257
;***** Secuencia 24 *****
e050i057:  btfss ADRESH,0
          goto e050i053
          goto e054i057
;***** Secuencia 25 *****
e390i397:  btfss ADRESH,0
          goto e390i393
          goto e394i397
;***** Secuencia 26 *****
e190i197:  btfss ADRESH,0
          goto e190i193
          goto e194i197
;***** Secuencia 27 *****
e290i297:  btfss ADRESH,0
          goto e290i293
          goto e294i297
;***** Secuencia 28 *****
e090i097:  btfss ADRESH,0
          goto e090i093
          goto e094i097
;***** Secuencia 29 *****
e310i317:  btfss ADRESH,0
          goto e310i313
          goto e314i317
;***** Secuencia 30 *****
e110i117:  btfss ADRESH,0
          goto e110i113
          goto e114i117
;***** Secuencia 31 *****
e210i217:  btfss ADRESH,0
          goto e210i213
          goto e214i217
```

```

;***** Secuencia 32 *****
e010i017:  btfss ADRESH,0
           goto  e010i013
           goto  e014i017
;***** Secuencia 33 *****
e3E0i3E7:  btfss ADRESH,0
           goto  e3E0i3E3
           goto  e3E4i3E7
;***** Secuencia 34 *****
e1E0i1E7:  btfss ADRESH,0
           goto  e1E0i1E3
           goto  e1E4i1E7

;***** Secuencia 35 *****
e2E0i2E7:  btfss ADRESH,0
           goto  e2E0i2E3
           goto  e2E4i2E7
;***** Secuencia 36 *****
e0E0i0E7:  btfss ADRESH,0
           goto  e0E0i0E3
           goto  e0E4i0E7
;***** Secuencia 37 *****
e360i367:  btfss ADRESH,0
           goto  e360i363
           goto  e364i367
;***** Secuencia 38 *****
e160i167:  btfss ADRESH,0
           goto  e160i163
           goto  e164i167
;***** Secuencia 39 *****
e260i267:  btfss ADRESH,0
           goto  e260i263
           goto  e264i267
;***** Secuencia 40 *****
e060i067:  btfss ADRESH,0
           goto  e060i063
           goto  e064i067
;***** Secuencia 41 *****
e3A0i3A7:  btfss ADRESH,0
           goto  e3A0i3A3
           goto  e3A4i3A7
;***** Secuencia 42 *****
e1A0i1A7:  btfss ADRESH,0
           goto  e1A0i1A3
           goto  e1A4i1A7
;***** Secuencia 43 *****
e2A0i2A7:  btfss ADRESH,0
           goto  e2A0i2A3
           goto  e2A4i2A7
;***** Secuencia 44 *****
e0A0i0A7:  btfss ADRESH,0
           goto  e0A0i0A3
           goto  e0A4i0A7
;***** Secuencia 45 *****
e320i327:  btfss ADRESH,0
           goto  e320i323
           goto  e324i327
;***** Secuencia 46 *****
e120i127:  btfss ADRESH,0
           goto  e120i123
           goto  e124i127

```

```
;***** Secuencia 47 *****
e220i227:  btfss ADRESH,0
           goto  e220i223
           goto  e224i227
;***** Secuencia 48 *****
e020i027:  btfss ADRESH,0
           goto  e020i023
           goto  e024i027
;***** Secuencia 49 *****
e3C0i3C7:  btfss ADRESH,0
           goto  e3C0i3C3
           goto  e3C4i3C7

;***** Secuencia 50 *****
e1C0i1C7:  btfss ADRESH,0
           goto  e1C0i1C3
           goto  e1C4i1C7
;***** Secuencia 51 *****
e2C0i2C7:  btfss ADRESH,0
           goto  e2C0i2C3
           goto  e2C4i2C7
;***** Secuencia 52 *****
e0C0i0C7:  btfss ADRESH,0
           goto  e0C0i0C3
           goto  e0C4i0C7
;***** Secuencia 53 *****
e340i347:  btfss ADRESH,0
           goto  e340i343
           goto  e344i347
;***** Secuencia 54 *****
e140i147:  btfss ADRESH,0
           goto  e140i143
           goto  e144i147
;***** Secuencia 55 *****
e240i247:  btfss ADRESH,0
           goto  e240i243
           goto  e244i247
;***** Secuencia 56 *****
e040i047:  btfss ADRESH,0
           goto  e040i043
           goto  e044i047
;***** Secuencia 57 *****
e380i387:  btfss ADRESH,0
           goto  e380i383
           goto  e384i387
;***** Secuencia 58 *****
e180i187:  btfss ADRESH,0
           goto  e180i183
           goto  e184i187
;***** Secuencia 59 *****
e280i287:  btfss ADRESH,0
           goto  e280i283
           goto  e284i287
;***** Secuencia 60 *****
e080i087:  btfss ADRESH,0
           goto  e080i083
           goto  e084i087
;***** Secuencia 61 *****
e300i307:  btfss ADRESH,0
           goto  e300i303
           goto  e304i307
```

```

;***** Secuencia 62 *****
e100i107:  btfss ADRESH,0
           goto  e100i103
           goto  e104i107
;***** Secuencia 63 *****
e200i207:  btfss ADRESH,0
           goto  e200i203
           goto  e204i207
;***** Secuencia 64 *****
e000i007:  btfss ADRESH,0
           goto  e000i003
           goto  e004i007

;*****
;*****SALIDAS*****
;*****
;*****
;**** FUNCIONES DE PERTINENCIA P Y PM (+0.4 V > ERROR > +0.3 V) *****
;*****
;En este caso el valor de salida es inmediato, ya que las salidas de
;estas FUNCIONES son constantes, sus valores pueden ser 255 (2,4 A)
;error 100% N a 170 (1,6 A) 100% NM.

e000i003:  movlw d'255'
           goto  fiprogramalop
e004i007:  movlw d'255'
           goto  fiprogramalop
e008i00B:  movlw d'255'
           goto  fiprogramalop
e00Ci00F:  movlw d'255'
           goto  fiprogramalop
e010i013:  movlw d'255'
           goto  fiprogramalop
e014i017:  movlw d'255'
           goto  fiprogramalop
e018i01B:  movlw d'255'
           goto  fiprogramalop
e01Ci01F:  movlw d'255'
           goto  fiprogramalop
e020i023:  movlw d'255'
           goto  fiprogramalop
e024i027:  movlw d'255'
           goto  fiprogramalop
e028i02B:  movlw d'255'
           goto  fiprogramalop
e02Ci02F:  movlw d'255'
           goto  fiprogramalop
e030i033:  movlw d'255'
           goto  fiprogramalop
e034i037:  movlw d'245'
           goto  fiprogramalop
e038i03B:  movlw d'235'
           goto  fiprogramalop
e03Ci03F:  movlw d'225'
           goto  fiprogramalop
e040i043:  movlw d'215'
           goto  fiprogramalop
e044i047:  movlw d'210'
           goto  fiprogramalop
e048i04B:  movlw d'207'
           goto  fiprogramalop

```

```

e04Ci04F:  movlw d'205'
           goto  fiprogramalop
e050i053:  movlw d'202'
           goto  fiprogramalop
e054i057:  movlw d'199'
           goto  fiprogramalop
e058i05B:  movlw d'197'
           goto  fiprogramalop
e05Ci05F:  movlw d'194'
           goto  fiprogramalop
e060i063:  movlw d'191'
           goto  fiprogramalop
e064i067:  movlw d'189'
           goto  fiprogramalop
e068i06B:  movlw d'186'
           goto  fiprogramalop
e06Ci06F:  movlw d'183'
           goto  fiprogramalop
e070i073:  movlw d'181'
           goto  fiprogramalop
e074i077:  movlw d'178'
           goto  fiprogramalop
e078i07B:  movlw d'175'
           goto  fiprogramalop
e07Ci07F:  movlw d'173'
           goto  fiprogramalop
e080i083:  movlw d'170'
           goto  fiprogramalop

;*****
;*** FUNCIONES DE PERTENENCIA PM Yy PS (+0.3 V > ERROR > +0.2 V)*****
;*****
;En caso del conjunto PM el valor de salida es inmediato a quest cuyo
;valor puede ir de de 165 ( $\cong 1.6$  A) a 0 (= 0 A), dependiendo del grado
;de pertenencia que tenga. El valor de la salida en el caso del
;conjunto borroso PS lo calcula el programa utilizando las tablas del
;K*error i del grado de pertenencia GRPERT . Al final del algoritmo se
;suman las salidas de los dos conjuntos borrosos.

e084i087:  movlw d'165'           ;valor correspondiente al
           movwf SALIDA_ANT2     ;conjunto borroso PM con su %
                                   ;aplicado.
           movlw d'23'           ;Valor correspondiente a la
                                   ;tabla K*error del conjunto ;
                                   ;borroso PS.
           call  rufercalsum      ;Calculamos la salida según
                                   ; el conjunto borroso PS.
           movlw 0x01            ;% que se aplicará
           movwf GRPERT          ;al conjunto PS 97%.
           goto  tantgeneral      ;Saltamos y realizamos la
                                   ; suma de las dos salidas,
                                   ;según PM i PS con el % ya
                                   ;aplicado.

e088i08B:  movlw d'160'           ;valor correspondiente al
           movwf SALIDA_ANT2     ;conjunto borroso PM con el %
                                   ;aplicado.
           movlw d'23'           ;Valor correspondiente a la
                                   ;tabla K*error del conjunto
                                   ;borroso PS.
           call  rufercalsum      ;Calculamos la salida según

```

```
movlw 0x02
movwf GRPERT
goto tantgeneral
;el conjunto borroso PS.
;% que se aplicará
;al conjunto PS 93%.
;Saltamos y realizamos la
;suma de las dos salidas,
;según PM i PS con el % ya
;aplicado.

e08Ci08F: movlw d'154'
movwf SALIDA_ANT2
movlw d'22'
call rufercalsum
movlw 0x03
movwf GRPERT
goto tantgeneral
; % que se aplicará
;al conjunto PS 91%.

e090i093: movlw d'149'
movwf SALIDA_ANT2
movlw d'22'
call rufercalsum
movlw 0x04
movwf GRPERT
goto tantgeneral
;% que se aplicará
;al conjunto PS 88%.

e094i097: movlw d'143'
movwf SALIDA_ANT2
movlw d'22'
call rufercalsum
movlw 0x05
movwf GRPERT
goto tantgeneral
;% que se aplicará
;al conjunto PS 85%.

e098i09B: movlw d'138'
movwf SALIDA_ANT2
movlw d'22'
call rufercalsum
movlw 0x06
movwf GRPERT
goto tantgeneral
;% que se aplicará
;al conjunto PS 82%.

e09Ci09F: movlw d'132'
movwf SALIDA_ANT2
movlw d'21'
call rufercalsum
movlw 0x07
movwf GRPERT
goto tantgeneral
;% que se aplicará
;al conjunto PS 79%.

e0A0i0A3: movlw d'127'
movwf SALIDA_ANT2
movlw d'21'
call rufercalsum
movlw 0x08
movwf GRPERT
goto tantgeneral
;% que se aplicará
;al conjunto PS 75%.

e0A4i0A7: movlw d'121'
movwf SALIDA_ANT2
movlw d'21'
call rufercalsum
movlw 0x09
movwf GRPERT
goto tantgeneral
;% que se aplicará
;al conjunto PS 72%.

e0A8i0AB: movlw d'116'
movwf SALIDA_ANT2
movlw d'21'
call rufercalsum
movlw 0x0A
;% que se aplicará
```

```
movwf GRPERT ;al conjunto PS 69%.
goto tantgeneral
e0ACi0AF: movlw d'110'
movwf SALIDA_ANT2
movlw d'20'
call rufercalsum
movlw 0x0B ;% que se aplicará
movwf GRPERT ;al conjunto PS 66%.
goto tantgeneral
e0B0i0B3: movlw d'105'
movwf SALIDA_ANT2
movlw d'20'
call rufercalsum
movlw 0x0C ;% que se aplicará
movwf GRPERT ;al conjunto PS 63%.
goto tantgeneral
e0B4i0B7: movlw d'100'
movwf SALIDA_ANT2
movlw d'20'
call rufercalsum
movlw 0x0D ;% que se aplicará
movwf GRPERT ;al conjunto PS 60%.
goto tantgeneral
e0B8i0BB: movlw d'94'
movwf SALIDA_ANT2
movlw d'20'
call rufercalsum
movlw 0x0E ;% que se aplicará
movwf GRPERT ;al conjunto PS 56%.
goto tantgeneral
e0BCi0BF: movlw d'89'
movwf SALIDA_ANT2
movlw d'19'
call rufercalsum
movlw 0x0F ;% que se aplicará
movwf GRPERT ;al conjunto PS 53%.
goto tantgeneral
e0C0i0C3: movlw d'83'
movwf SALIDA_ANT2
movlw d'19'
call rufercalsum
movlw 0x10 ;% que se aplicará
movwf GRPERT ;al conjunto PS 50%.
goto tantgeneral
e0C4i0C7: movlw d'78'
movwf SALIDA_ANT2
movlw d'19'
call rufercalsum
movlw 0x11 ;% que se aplicará
movwf GRPERT ;al conjunto PS 47%.
goto tantgeneral
e0C8i0CB: movlw d'72'
movwf SALIDA_ANT2
movlw d'19'
call rufercalsum
movlw 0x12 ;% que se aplicará
movwf GRPERT ;al conjunto PS 44%.
goto tantgeneral
e0CCi0CF: movlw d'67'
movwf SALIDA_ANT2
movlw d'18'
```

```
call rufercalsum
movlw 0x13 ;% que se aplicará
movwf GRPERT ;al conjunto PS 41%.
goto tantgeneral
e0D0i0D3: movlw d'61'
movwf SALIDA_ANT2
movlw d'18'
call rufercalsum
movlw 0x14 ;% que se aplicará
movwf GRPERT ;al conjunto PS 38%.
goto tantgeneral
e0D4i0D7: movlw d'56'
movwf SALIDA_ANT2
movlw d'18'
call rufercalsum
movlw 0x15 ;% que se aplicará
movwf GRPERT ;al conjunto PS 34%.
goto tantgeneral
e0D8i0DB: movlw d'50'
movwf SALIDA_ANT2
movlw d'18'
call rufercalsum
movlw 0x16 ;% que se aplicará
movwf GRPERT ;al conjunto PS 31%.
goto tantgeneral
e0DCi0DF: movlw d'45'
movwf SALIDA_ANT2
movlw d'17'
call rufercalsum
movlw 0x17 ;% que se aplicará
movwf GRPERT ;al conjunto PS 28%.
goto tantgeneral
e0E0i0E3: movlw d'39'
movwf SALIDA_ANT2
movlw d'17'
call rufercalsum
movlw 0x18 ;% que se aplicará
movwf GRPERT ;al conjunto PS 25%.
goto tantgeneral
e0E4i0E7: movlw d'34'
movwf SALIDA_ANT2
movlw d'17'
call rufercalsum
movlw 0x19 ;% que se aplicará
movwf GRPERT ;al conjunto PS 23%.
goto tantgeneral
e0E8i0EB: movlw d'28'
movwf SALIDA_ANT2
movlw d'17'
call rufercalsum
movlw 0x1A ;% que se aplicará
movwf GRPERT ;al conjunto PS 19%.
goto tantgeneral
e0ECi0EF: movlw d'23'
movwf SALIDA_ANT2
movlw d'16'
call rufercalsum
movlw 0x1B ;% que se aplicará
movwf GRPERT ;al conjunto PS 16%.
goto tantgeneral
e0F0i0F3: movlw d'17'
```



```

        movwf SALIDA_ANT2
        movlw d'16'
        call rufercalsum
        movlw 0x1C                                ;% que se aplicará
        movwf GRPERT                              ;al conjunto PS 12%.
        goto tantgeneral
e0F4i0F7:  movlw d'12'
        movwf SALIDA_ANT2
        movlw d'16'
        call rufercalsum
        movlw 0x1D                                ;% que se aplicará
        movwf GRPERT                              ;al conjunto PS 09%.
        goto tantgeneral
e0F8i0FB:  movlw d'07'
        movwf SALIDA_ANT2
        movlw d'16'
        call rufercalsum
        movlw 0x1E                                ;% que se aplicará
        movwf GRPERT                              ;al conjunto PS 06%.
        goto tantgeneral
e0FCi0FF:  movlw d'02'
        movwf SALIDA_ANT2
        movlw d'15'
        call rufercalsum
        movlw 0x1F                                ;% que se aplicará
        movwf GRPERT                              ;al conjunto PS 03%.
        goto tantgeneral

;*****
;***** FUNCIÓN DE PERTINENCIA PS (+0.2 V > ERROR > +0.0125 V) *****
;*****
;en este caso el valor de salida sólo depende de un conjunto (PS).

e100i103:
e104i107:
e108i10B:
        e10Ci10F:  movlw d'15'                    ;Valor correspondiente a la
                                                ;tabla K*error del conjunto
                                                ;borroso PS.
        call rufercalsum                        ;Calculamos la salida según
                                                ;el conjunto borroso PS.
        goto fiprogramalop                     ;saltamos al final del
                                                ; programa,el valor de salida
                                                ;actualitzado ya está en el
                                                ;registro W

e110i113:
e114i117:
e118i11B:
e11Ci11F:  movlw d'14'
        call rufercalsum
        goto fiprogramalop

e120i123:
e124i127:
e128i12B:
e12Ci12F:  movlw d'13'
        call rufercalsum
        goto fiprogramalop

e130i133:
e134i137:
e138i13B:
e13Ci13F:

```

```
e140i143:    movlw d'12'  
            call  rufercalsum  
            goto  fiprogramalop  
  
e144i147:  
e148i14B:  
e14Ci14F:  
e150i153:    movlw d'11'  
            call  rufercalsum  
            goto  fiprogramalop  
  
e154i157:  
e158i15B:  
e15Ci15F:    movlw d'10'  
            call  rufercalsum  
            goto  fiprogramalop  
  
e160i163:  
e164i167:  
e168i16B:  
e16Ci16F:    movlw d'09'  
            call  rufercalsum  
            goto  fiprogramalop  
  
e170i173:  
e174i177:  
e178i17B:  
e17Ci17F:  
e180i183:    movlw d'08'  
            call  rufercalsum  
            goto  fiprogramalop  
  
e184i187:  
e188i18B:  
e18Ci18F:  
e190i193:    movlw d'07'  
            call  rufercalsum  
            goto  fiprogramalop  
  
e194i197:  
e198i19B:  
e19Ci19F:  
e1A0i1A3:    movlw d'06'  
            call  rufercalsum  
            goto  fiprogramalop  
  
e1A4i1A7:  
e1A8i1AB:  
e1ACi1AF:  
e1B0i1B3:    movlw d'05'  
            call  rufercalsum  
            goto  fiprogramalop  
  
e1B4i1B7:  
e1B8i1BB:  
e1BCi1BF:  
e1C0i1C3:    movlw d'04'  
            call  rufercalsum  
            goto  fiprogramalop  
  
e1C4i1C7:  
e1C8i1CB:  
e1CCi1CF:  
e1D0i1D3:    movlw d'03'  
            call  rufercalsum  
            goto  fiprogramalop  
  
e1D4i1D7:  
e1D8i1DB:  
e1DCi1DF:  
e1E0i1E3:    movlw d'02'
```

```

        call  rufercalsum
        goto  fiprogramalop
e1E4i1E7:
e1E8i1EB:
e1ECi1EF:  movlw d'01'
           call  rufercalsum
           goto  fiprogramalop

;*****
;*****
;*****
;*****
;Testeamos los 2 bits de menos peso, para obtener; una mejor
;resolución del error de entrada, ya que en este caso ya estamos cerca
;del punto de equilibrio (ERROR  $\cong$  0).

;EN EL DIAGRAMA DE CORRESPONDE AL ESTADO E
;*****
;*****

e1F0i1F3:  bsf    STATUS,RP0
           btfss ADRESL,7
           goto  e1F0i1F1
           btfss ADRESL,6
           goto  valor1F2
           goto  valor1F3
e1F0i1F1:  btfss ADRESL,6
           goto  valor1F0
           goto  valor1F1

e1F4i1F7:  bsf    STATUS,RP0
           btfss ADRESL,7
           goto  e1F4i1F5
           btfss ADRESL,6
           goto  valor1F6
           goto  valor1F7
e1F4i1F5:  btfss ADRESL,6
           goto  valor1F4
           goto  valor1F5

e1F8i1FB:  bsf    STATUS,RP0
           btfss ADRESL,7
           goto  e1F8i1F9
           btfss ADRESL,6
           goto  valor1FA
           goto  valor1FB
e1F8i1F9:  btfss ADRESL,6
           goto  valor1F8
           goto  valor1F9

e1FCi1FF:  bsf    STATUS,RP0
           btfss ADRESL,7
           goto  e1FCi1FD
           btfss ADRESL,6
           goto  valor1FE
           goto  valor1FF
e1FCi1FD:  btfss ADRESL,6
           goto  valor1FC
           goto  valor1FD

```

```

e200i203:  bsf    STATUS,RP0
           btfss  ADRESL,7
           goto   e200i201
           btfss  ADRESL,6
           goto   valor202
           goto   valor203
e200i201:  btfss  ADRESL,6
           goto   valor200
           goto   valor201

e204i207:  bsf    STATUS,RP0
           btfss  ADRESL,7
           goto   e204i205
           btfss  ADRESL,6
           goto   valor206
           goto   valor207
e204i205:  btfss  ADRESL,6
           goto   valor204
           goto   valor205

e208i20B:  bsf    STATUS,RP0
           btfss  ADRESL,7
           goto   e208i209
           btfss  ADRESL,6
           goto   valor20A
           goto   valor20B
e208i209:  btfss  ADRESL,6
           goto   valor208
           goto   valor209

e20Ci20F:  bsf    STATUS,RP0
           btfss  ADRESL,7
           goto   e20Ci20D
           btfss  ADRESL,6
           goto   valor20E
           goto   valor20F
e20Ci20D:  btfss  ADRESL,6
           goto   valor20C
           goto   valor20D

```

```

;*****
;***** Subrutinas aplicadas en el programa para el cálculo *****
;***** de las salidas que implementan PI's.*****
;*****

```

```

;*****
;***** Subrutina RUFERCALRES *****
;*****

```

```

; Esta rutina calcula la salida en el caso que el error sea negativo.
;Guarda la salida en el registro W y en el registro OUT1.

```

```

;***** Salida = - K*error ± Kc*cerror + Iref(anterior) *****

```

rufercalres:

```

    subwf  PORTB,W           ;En esta operación el registro W
                             ; posee el siguiente valor:
                             ;Iref(anterior)- K*error
    btfss  STATUS,C         ;miramos el bit de carry, si la

```

```

        clrw                ; resta es negativa el registro W=0

        btfsc SIGNECERROR,0 ;miramos el signo de la variable
        goto  restarr      ; CERROR.Si es negativo saltamos.
        addwf CERROR_NOR7,W ;sino sumamos al registro W el
                           ; valor de  $K_c * cerror$ , y el registro
                           ; W tendrá el valor:-
                           ;  $K * error + K_c * cerror + Iref(ant)$ .

        btfsc STATUS,C     ;Miramos el bit de carry, y si la
                           ; suma anterior ha desbordado el
        movlw 0xFF         ; registro. W= valor máximo de
                           ; salida 0xFF (2,4 A).

        goto  fifercalculsr ;Saltamos al final de la subrutina.
restarr: movwf AUX         ;AUX =  $Iref(anterior) - K * error$ 
        movf  CERROR_NOR7,W
        subwf AUX,W        ;W =  $-K * error - K_c * cerror + Iref(ant)$ .
        btfss STATUS,C     ;Miramos el bit de carry, y si la
                           ;resta anterior ha desbordado el
        movlw d'11'       ;registro. W = 0.1 A, un valor bajo
                           ; de salida, pero no nulo, se ha
                           ; optado por poner un valor mayor
                           ; que 0 para evitar que el
                           ; convertidor entre en modo
                           ; discontinuo.

fifercalculsr:
        movwf OUT1        ;Copiamos el valor de registro W en
                           ; la posición de memoria OUT1, que
                           ; posteriormente será utilizada para
                           ; el cálculo del % si es necesario.

        return

;*****
;***** Subrutina RUFERCALSUM *****
;*****
;Esta rutina calcula la salida cuando el error de entrada es positivo,
; y guarda la salida en los registros W y OUT1. Es igual que la anterior
; exceptuando el valor de K*error, que en este caso es positivo.
;***** Salida = + K*error ± Kc*ccerror + Iref(anterior) *****

rufercalsum:
        addwf PORTB,W     ;En esta operación el registro W
                           ; tiene el siguiente valor:
                           ;  $Iref(anterior) + K * error$ 
        btfsc STATUS,C     ;Miramos el bit de carry, y si la
                           ; suma anterior ha desbordado el
        movlw 0xFF         ; registro. W=valor máximo de salida
                           ; 0xFF (2,4 A).

        btfsc SIGNECERROR,0 ;....
        goto  restars      ;.....
        addwf CERROR_NOR7,W
        btfsc STATUS,C
        movlw 0xFF
        goto  fifercalculss

restars: movwf AUX
        movf  CERROR_NOR7,W
        subwf AUX,W
        btfss STATUS,C

```

```

movlw d'11'

fifercalculss:
    movwf OUT1
    return

;*****
;***** Subrutina ZERO *****
;*****
;Esta rutina calcula la salida cuando el error de entrada sólo
;pertenece al conjunto Z. La variable K*error es tan pequeña, que se
;ha igualado a zero. Por esto no se evalua.
;***** Salida = ± Kc*cerror + Iref(anterior) *****

zero:    movf  CERROR_NOR2,W      ;W = Kc*cerror.
         btfsc SIGNECERROR,0    ;Miramos el signo del CERROR,
         goto restarz          ;Si es negativo saltamos a restarz
         addwf PORTB,W         ;sino sumamos, W = +Kc*cerror+Iref.
         btfsc STATUS,C        ;Si se desborda el valor por la
                                ;suma anterior W= 255, es decir
                                ;Iref = 2,4 A.
         movlw 0xFF            ;saltamos al final de la rutina.
         goto fifercalculsz

restarz: subwf PORTB,W          ;W = -Kc*cerror+Iref
         btfss STATUS,C        ;Si se desborda el valor por la
                                ;resta anterior W= 11, es decir
                                ;Iref= 0.1 A
         movlw d'11'

fifercalculsz:
         movwf OUT2            ;Copiamos el valor de registro W en
                                ;la posición de memoria OUT2, ;que
                                ;posteriormente será usada para el
                                ;cálculo del % si es necesario

         return

;*****
;***** Subrutina adap *****
;*****
;Esta rutina multiplica por 4 los valores guardados del
;CERROR(multiplicados por sus ctes (*2 y*7)).De esta manera se obtiene
;más resolución en la salida. En la rutina TANTPERCENTZERO, y después
;de realizar los cálculos pertinentes, estos valores serán divididos
;por 4.
;EN EL DIAGRAMA DE CORRESPONDE AL ESTADO (H)

adap:   movf  PORTB,W
         movwf PORTB_ANT      ;Iref(k-1) = PORTB_ANT.

         bcf   STATUS,C
         rlf   CERROR_NOR2,F  ;Multiplicamos por 2 el valor de
                                ;de la variable CERROR_NOR2
         bcf   STATUS,C
         rlf   CERROR_NOR2,w  ;volvemos a multiplicar por 2, el
                                ;valor de la variable CERROR_NOR2.
                                ;En total multiplicamos por 4.
         movwf OUT2          ;Guardamos el valor multiplicado en
                                ;la posición de memoria OUT2, que
                                ;posteriormente será utilizada
                                ;para calcular el %.

         ;Lo mismo para la variable CERROR_NOR7

         bcf   STATUS,C

```

```

rlf  CERROR_NOR7,F      ;*2
bcf  STATUS,C
rlf  CERROR_NOR7,w      ;*4
movwf OUT1
return

;*****
;** FUNCIONES DE PERTENENCIA PS i Z (+0.0125 V > ERROR > +0.00625 V) *
;*****
;El valor de salida en el caso del conjunto PS y Z es un PI. El valor
;final de la salida será la suma aplicando el % correspondiente.
;Salida del conjunto borroso PS=Iref(anterior)+K1*CERROR+K2*ERROR)*x%
;*** Salida del conjunto borroso Z = (Iref(anterior)+K3*CERROR)*y% ***
;***** x + y = 1 *****
; Salida final = Iref(anterior)+(K1*CERROR+K2*ERROR)*x%+K3*CERROR*y% *

valor1F0:
valor1F1:  bcf  STATUS,RP0      ;cambio de banco 01-->00
           call adap
           incf  PORTB_ANT,F      ;esta instrucción corresponde a
                                   ;K2*ERROR*x% (error positivo),
                                   ;en este momento del programa la
                                   ;salida es Iref(ant)+(K2*ERROR)*x%
                                   ;En la rutina tantzero se realizan
                                   ;los cálculos necesarios según la
                                   ;variable CERROR.
           movlw 0xE2           ;% que se aplicará al
                                   ;conjunto PS (valor E) 87% y al
                                   ;conjunto Z(valor 2) 13%.
           movwf GRPERT
           goto  tantzero      ;Saltamos a esta posición para
                                   ;el cálculo de la salida final.

valor1F2:  bcf  STATUS,RP0      ;cambio de banco 01-->00
           call adap
           incf  PORTB_ANT,F      ;esta instrucción corresponde a
                                   ;K2*ERROR*x% (error positivo),
                                   ;en este momento del programa la
                                   ;salida es Iref(ant)+(K2*ERROR)*x%
                                   ;En la rutina tantzero se realizan
                                   ;los cálculos necesarios según la
                                   ;variable CERROR.
           movlw 0xC4           ;% que se aplicará al
                                   ;conjunto PS (valor C) 75% y al
                                   ;conjunto Z(valor 4) 25%.
           movwf GRPERT
           goto  tantzero      ;Saltamos para el cálculo de la
                                   ;salida final.

valor1F3:  bcf  STATUS,RP0      ;cambio de banco 01-->00
           call adap
           incf  PORTB_ANT,F      ;esta instrucción corresponde a
                                   ;K2*ERROR*x% (error positivo),
                                   ;en este momento del programa la
                                   ;salida es Iref(ant)+(K2*ERROR)*x%
                                   ;En la rutina tantzero se realizan
                                   ;los cálculos necesarios según la
                                   ;variable CERROR.
           movlw 0xA6           ;% que se aplicará al
                                   ;conjunto PS (valor A) 62% y al
                                   ;conjunto Z(valor 6) 38%.
           movwf GRPERT
           goto  tantzero      ;Saltamos para el cálculo de la
                                   ;salida final.

valor1F4:  bcf  STATUS,RP0      ;cambio de banco 01-->00

```

```

call adap
incf PORTB_ANT,F
movlw 0x88                ;% que se aplicará al
                          ;conjunto PS (valor 8) 50% y al
                          ;conjunto Z(valor 8) 50%.

movwf GRPERT
goto tantzero
valor1F5: bcf STATUS,RP0    ;cambio de banco 01-->00
call adap
movlw 0x6A                ;% que se aplicará al
                          ;conjunto PS (valor 6) 38% y al
                          ;conjunto Z(valor A) 62%.

movwf GRPERT
goto tantzero

valor1F6: bcf STATUS,RP0    ;cambio de banco 01-->00
call adap
movlw 0x4C                ;% que se aplicará al
                          ;conjunto PS (valor 4) 25% y al
                          ;conjunto Z(valor C) 75%.

movwf GRPERT
goto tantzero

valor1F7: bcf STATUS,RP0    ;cambio de banco 01-->00
call adap
movlw 0x2E                ;% que se aplicará al
                          ;conjunto PS (valor 2) 12% y al
                          ;conjunto Z(valor E) 88%.

movwf GRPERT
goto tantzero            ;Saltamos para el cálculo de la
                          ;salida final

;*****
;***** FUNCIÓN DE PERTENENCIA Z (+0.00625 V > ERROR > -0.00625 V) ****
;*****
;***** Salida = ± Kc*cerror + ref(anterior)*****

valor1F8:
valor1F9:
valor1FA:
valor1FB:
valor1FC:
valor1FD:
valor1FE:
valor1FF:
valor200:
valor201:
valor202:
valor203:
valor204:
valor205:
valor206:
valor207: bcf STATUS,RP0    ;cambio de banco 01-->00
call zero                ;llamamos a la rutina que realizará
                          ;los cálculos.

goto fiprogramalop      ;el valor de salida está guardado
                          ;en el registro W.

```



```

;*****
;* FUNCIONES DE PERTINENCIA Z i NS (-0.00625 V < ERROR < -0.0125 V) **
;*****
;Esta zona es igual que la PS y Z, pero cambiando el signo de K*ERROR.
;Salida del conjunto borroso NS=(Iref(anterior)+K1*CERROR+K2*ERROR)*x%
;*** Salida del conjunto borroso Z = (Iref(anterior)+K3*CERROR)*y% ***
;***** x + y = 1 *****
;*Salida final = Iref(anterior)+(K1*CERROR+K2*ERROR)*x%+K3*CERROR*y% *

valor208:   bcf    STATUS,RP0      ;cambio de banco 01-->00
            call   adap
            movlw  0x2E          ;% que se aplicará al
                                   ;conjunto Z (valor E) 87% y al
                                   ;conjunto NS(valor 2) 13%.

            movwf  GRPERT
            goto  tantzero      ;Saltamos para el cálculo de la

valor209:   bcf    STATUS,RP0      ;cambio de banco 01-->00
            call   adap
            movlw  0x4C          ;75% Z, 25% NM.
            movwf  GRPERT
            goto  tantzero

valor20A:   bcf    STATUS,RP0      ;cambio de banco 01-->00
            call   adap
            movlw  0x6A          ;62% Z, 38% NM.
            movwf  GRPERT
            goto  tantzero

valor20B:   bcf    STATUS,RP0      ;cambio de banco 01-->00
            call   adap
            decf   PORTB_ANT,F    ;esta instrucción corresponde a
                                   ;K2*ERROR*x% (error NEGATIU),
                                   ;en este momento del programa la
                                   ;salida es Iref(ant)-(K2*ERROR)*x%
                                   ;En la rutina tantzero se realizan
                                   ;los cálculos necesarios según la
                                   ;variable CERROR.

            movlw  0x88          ;50% Z, 50% NM.
            movwf  GRPERT
            goto  tantzero

valor20C:   bcf    STATUS,RP0      ;cambio de banco 01-->00
            call   adap
            decf   PORTB_ANT,F
            movlw  0xA6          ;37% Z, 63% NM.
            movwf  GRPERT
            goto  tantzero

valor20D:   bcf    STATUS,RP0      ;cambio de banco 01-->00
            call   adap
            decf   PORTB_ANT,F
            movlw  0xC4          ;25% Z, 75% NM.
            movwf  GRPERT
            goto  tantzero

valor20E:
valor20F:   bcf    STATUS,RP0      ;cambio de banco 01-->00
            call   adap
            decf   PORTB_ANT,F
            movlw  0xE2          ;12% Z, 88% NM.
            movwf  GRPERT
            goto  tantzero

```

```

;*****
;***** FUNCIÓN DE PERTINENCIA NS (-0.2 V < ERROR < -0.0125 V) *****
;*****
;En este caso el valor de salida sólo depende de un conjunto (PS).

```

```

e210i213:
e214i217:
e218i21B:   movlw d'01'           ;Valor correspondiente a la
                                           ;tabla K*error del conjunto
                                           ;borroso NS.
                                           ;Calculamos la salida según
                                           ;el conjunto borroso NS.
           call  rufercalres      ;Saltamos al final del
                                           ;programa,.El valor de
                                           ;salida actualitzado ya está
                                           ;en el registro W.

e21Ci21F:
e220i223:
e224i227:
e228i22B:   movlw d'02'           ;Valor correspondiente a la
                                           ;tabla K*error del conjunto
                                           ;borroso NS.
                                           ;Calculamos la salida según
                                           ;el conjunto borroso NS.
           call  rufercalres      ;Saltamos al final del
                                           ;programa. El valor de salida
                                           ;actualizado ya está en el
                                           ;registro W.

e22Ci22F:
e230i233:
e234i237:
e238i23B:   movlw d'03'           ;.....
           call  rufercalres      ;.....
           goto  fiprogramalop    ;....

e23Ci23F:
e240i243:
e244i247:
e248i24B:   movlw d'04'
           call  rufercalres
           goto  fiprogramalop

e24Ci24F:
e250i253:
e254i257:
e258i25B:   movlw d'05'
           call  rufercalres
           goto  fiprogramalop

e25Ci25F:
e260i263:
e264i267:
e268i26B:   movlw d'06'
           call  rufercalres
           goto  fiprogramalop

e26Ci26F:
e270i273:
e274i277:
e278i27B:   movlw d'07'
           call  rufercalres
           goto  fiprogramalop

e27Ci27F:

```

```
e280i283:
e284i287:
e288i28B:
e28Ci28F:    movlw d'08'
              call  rufercalres
              goto  fiprogramalop

e290i293:
e294i297:
e298i29B:
e29Ci29F:    movlw d'09'
              call  rufercalres
              goto  fiprogramalop

e2A0i2A3:
e2A4i2A7:
e2A8i2AB:
e2ACi2AF:    movlw d'10'
              call  rufercalres
              goto  fiprogramalop

e2B0i2B3:
e2B4i2B7:
e2B8i2BB:
e2BCi2BF:    movlw d'11'
              call  rufercalres
              goto  fiprogramalop

e2C0i2C3:
e2C4i2C7:
e2C8i2CB:
e2CCi2CF:    movlw d'12'
              call  rufercalres
              goto  fiprogramalop

e2D0i2D3:
e2D4i2D7:
e2D8i2DB:
e2DCi2DF:    movlw d'13'
              call  rufercalres
              goto  fiprogramalop

e2E0i2E3:
e2E4i2E7:
e2E8i2EB:
e2ECi2EF:
e2F0i2F3:    movlw d'14'
              call  rufercalres
              goto  fiprogramalop

e2F4i2F7:
e2F8i2FB:
e2FCi2FF:    movlw d'15'
              call  rufercalres
              goto  fiprogramalop
```

```

;*****
;*** FUNCIONES DE PERTINENCIA NS I NM (-0.3 V < ERROR < -0.2 V) *****
;*****
;El valor de salida en el caso del conjunto borroso NM es inmediato.
;Su rango de valores es de 107 ( $\cong 1$  A) a 0 (= 0 A).Dependiendo del
;grado de pertinencia que tenga. El valor de salida en el caso del
;conjunto borroso NS está calculado por el algoritmo. Para ello, se
;utilizan las tablas del K*error y del grado de pertenencia GRPERT
;explicadas en la memoria de cálculo. Al final del algoritme se suman
;las dos salidas de los dos conjunts borrosos.

e300i303:  movlw d'15'
           call  rufercalres
           goto  fiprogramalop

e304i307:  movlw d'03'
           movwf SALIDA_ANT2
           movlw d'16'
           call  rufercalres
           movlw 0x1F
           movwf GRPERT
           goto  tantgeneral

           ;Valor correspondiente al
           ;conjunto borroso NM,CON el %
           ;aplicado.
           ;Valor correspondiente a la
           ;tabla K*error del conjunto
           ;borroso NS.
           ;Calculamos la salida según
           ;el conjunto borroso NS.
           ;% que se aplicará
           ;al conjunto NS 97%.
           ;Saltamos y realizamos la
           ;suma de las dos salidas,
           ;según NM i PS con el % ya
           ;aplicado.

e308i30B:  movlw d'07'
           movwf SALIDA_ANT2
           movlw d'16'
           call  rufercalres
           movlw 0x1E
           movwf GRPERT
           goto  tantgeneral

           ;Valor correspondiente al
           ;conjunto borroso NM, con el
           ; % aplicado.
           ;Valor correspondiente a la
           ;tabla K*error del conjunto
           ;borroso NS.
           ;Calculamos la salida según
           ;el conjunto borroso NS.
           ;% que se aplicará
           ;al conjunto NS 93%
           ;Saltamos y realizamos la
           ; suma de las dos salidas,
           ;según NM i PS con el % ya
           ;aplicado.

e30Ci30F:  movlw d'10'
           movwf SALIDA_ANT2
           movlw d'16'
           call  rufercalres
           movlw 0x1D
           movwf GRPERT
           goto  tantgeneral

           ;% que se aplicará
           ;al conjunto NS 91%.

e310i313:  movlw d'14'
           movwf SALIDA_ANT2
           movlw d'16'
           call  rufercalres
           movlw 0x1C
           movwf GRPERT
           goto  tantgeneral

           ;% que se aplicará
           ;al conjunto NS 87%.

e314i317:  movlw d'17'
           movwf SALIDA_ANT2
           movlw d'17'

```

```

        call  rufercalres
        movlw 0x1B                               ;% que se aplicará
        movwf GRPERT                             ;al conjunto NS 84%.
        goto  tantgeneral
e318i31B: movlw d'21'
        movwf SALIDA_ANT2
        movlw d'17'
        call  rufercalres
        movlw 0x1A                               ;% que se aplicará
        movwf GRPERT                             ;al conjunto NS 81%.
        goto  tantgeneral
e31Ci31F: movlw d'24'
        movwf SALIDA_ANT2
        movlw d'17'
        call  rufercalres
        movlw 0x19                               ;% que se aplicará
        movwf GRPERT                             ;al conjunto NS 78%.
        goto  tantgeneral
e320i323: movlw d'27'
        movwf SALIDA_ANT2
        movlw d'17'
        call  rufercalres
        movlw 0x18                               ;% que se aplicará
        movwf GRPERT                             ;al conjunto NS 75%.
        goto  tantgeneral
e324i327: movlw d'31'
        movwf SALIDA_ANT2
        movlw d'18'
        call  rufercalres
        movlw 0x17                               ;% que se aplicará
        movwf GRPERT                             ;al conjunto NS 71%.
        goto  tantgeneral
e328i32B: movlw d'34'
        movwf SALIDA_ANT2
        movlw d'18'
        call  rufercalres
        movlw 0x16                               ;% que se aplicará
        movwf GRPERT                             ;al conjunto NS 68%.
        goto  tantgeneral
e32Ci32F: movlw d'37'
        movwf SALIDA_ANT2
        movlw d'18'
        call  rufercalres
        movlw 0x15                               ;% que se aplicará
        movwf GRPERT                             ;al conjunto NS 65%.
        goto  tantgeneral
e330i333: movlw d'41'
        movwf SALIDA_ANT2
        movlw d'18'
        call  rufercalres
        movlw 0x14                               ;% que se aplicará
        movwf GRPERT                             ;al conjunto NS 62%.
        goto  tantgeneral
e334i337: movlw d'44'
        movwf SALIDA_ANT2
        movlw d'19'
        call  rufercalres
        movlw 0x13                               ;% que se aplicará
        movwf GRPERT                             ;al conjunto NS 59%.
        goto  tantgeneral
e338i33B: movlw d'48'

```

```
movwf SALIDA_ANT2
movlw d'19'
call rufercalres
movlw 0x12                                ;% que se aplicará
movwf GRPERT                              ;al conjunto NS 56%.
goto tantgeneral
e33Ci33F: movlw d'51'
movwf SALIDA_ANT2
movlw d'19'
call rufercalres
movlw 0x11                                ;% que se aplicará
movwf GRPERT                              ;al conjunto NS 53%.
goto tantgeneral
e340i343: movlw d'55'
movwf SALIDA_ANT2
movlw d'19'
call rufercalres
movlw 0x10                                ;% que se aplicará
movwf GRPERT                              ;al conjunto NS 50%.
goto tantgeneral
e344i347: movlw d'58'
movwf SALIDA_ANT2
movlw d'20'
call rufercalres
movlw 0x0F                                ;% que se aplicará
movwf GRPERT                              ;al conjunto NS 47%.
goto tantgeneral
e348i34B: movlw d'62'
movwf SALIDA_ANT2
movlw d'20'
call rufercalres
movlw 0x0E                                ;% que se aplicará
movwf GRPERT                              ;al conjunto NS 44%.
goto tantgeneral
e34Ci34F: movlw d'65'
movwf SALIDA_ANT2
movlw d'20'
call rufercalres
movlw 0x0D                                ;% que se aplicará
movwf GRPERT                              ;al conjunto NS 41%.
goto tantgeneral
e350i353: movlw d'69'
movwf SALIDA_ANT2
movlw d'20'
call rufercalres
movlw 0x0C                                ;% que se aplicará
movwf GRPERT                              ;al conjunto NS 37%.
goto tantgeneral
e354i357: movlw d'72'
movwf SALIDA_ANT2
movlw d'20'
call rufercalres
movlw 0x0B                                ;% que se aplicará
movwf GRPERT                              ;al conjunto NS 35%.
goto tantgeneral
e358i35B: movlw d'76'
movwf SALIDA_ANT2
movlw d'21'
call rufercalres
movlw 0x0A                                ;% que se aplicará
movwf GRPERT                              ;al conjunto NS 31%.
```

```
e35Ci35F:   goto   tantgeneral
            movlw  d'79'
            movwf  SALIDA_ANT2
            movlw  d'21'
            call   rufercalres
            movlw  0x09                               ;% que se aplicará
            movwf  GRPERT                               ;al conjunto NS 28%.
            goto   tantgeneral
e360i363:   movlw  d'83'
            movwf  SALIDA_ANT2
            movlw  d'21'
            call   rufercalres
            movlw  0x08                               ;% que se aplicará
            movwf  GRPERT                               ;al conjunto NS 25%.
            goto   tantgeneral
e364i367:   movlw  d'86'
            movwf  SALIDA_ANT2
            movlw  d'21'
            call   rufercalres
            movlw  0x07                               ;% que se aplicará
            movwf  GRPERT                               ;al conjunto NS 21%.
            goto   tantgeneral
e368i36B:   movlw  d'90'
            movwf  SALIDA_ANT2
            movlw  d'22'
            call   rufercalres
            movlw  0x06                               ;% que se aplicará
            movwf  GRPERT                               ;al conjunto NS 18%.
            goto   tantgeneral
e36Ci36F:   movlw  d'93'
            movwf  SALIDA_ANT2
            movlw  d'22'
            call   rufercalres
            movlw  0x05                               ;% que se aplicará
            movwf  GRPERT                               ;al conjunto NS 15%.
            goto   tantgeneral
e370i373:   movlw  d'97'
            movwf  SALIDA_ANT2
            movlw  d'22'
            call   rufercalres
            movlw  0x04                               ;% que se aplicará
            movwf  GRPERT                               ;al conjunto NS 12%.
            goto   tantgeneral
e374i377:   movlw  d'100'
            movwf  SALIDA_ANT2
            movlw  d'22'
            call   rufercalres
            movlw  0x03                               ;% que se aplicará
            movwf  GRPERT                               ;al conjunto NS 09%.
            goto   tantgeneral
e378i37B:   movlw  d'104'
            movwf  SALIDA_ANT2
            movlw  d'23'
            call   rufercalres
            movlw  0x02                               ;% que se aplicará
            movwf  GRPERT                               ;al conjunto NS 06%.
            goto   tantgeneral
e37Ci37F:   movlw  d'107'
            movwf  SALIDA_ANT2
            movlw  d'23'
            call   rufercalres
```

```
        movlw 0x01                ;% que se aplicará
        movwf GRPERT              ;al conjunto NS 03%.
        goto tantgeneral
;*****
;*** FUNCIONES DE PERTINENCIA N i NM (-0.4 V < ERROR < -0.3 V) *****
;*****
;El valor de salida es inmediato, ya que los consecuentes son ctes. El
; rango de la salida es de 105 (1 A) error 100% N a 59 (0.4 A) 100% NM.

e380i383:  movlw d'105'
           goto fiprogramalop
e384i387:  movlw d'103'
           goto fiprogramalop
e388i38B:  movlw d'101'
           goto fiprogramalop
e38Ci38F:  movlw d'99'
           goto fiprogramalop
e390i393:  movlw d'97'
           goto fiprogramalop
e394i397:  movlw d'95'
           goto fiprogramalop
e398i39B:  movlw d'93'
           goto fiprogramalop
e39Ci39F:  movlw d'91'
           goto fiprogramalop
e3A0i3A3:  movlw d'89'
           goto fiprogramalop
e3A4i3A7:  movlw d'87'
           goto fiprogramalop
e3A8i3AB:  movlw d'85'
           goto fiprogramalop
e3ACi3AF:  movlw d'83'
           goto fiprogramalop
e3B0i3B3:  movlw d'81'
           goto fiprogramalop
e3B4i3B7:  movlw d'79'
           goto fiprogramalop
e3B8i3BB:  movlw d'77'
           goto fiprogramalop
e3BCi3BF:  movlw d'75'
           goto fiprogramalop
e3C0i3C3:  movlw d'73'
           goto fiprogramalop
e3C4i3C7:  movlw d'71'
           goto fiprogramalop
e3C8i3CB:  movlw d'69'
           goto fiprogramalop
e3CCi3CF:  movlw d'67'
           goto fiprogramalop
e3D0i3D3:  movlw d'65'
           goto fiprogramalop
e3D4i3D7:  movlw d'63'
           goto fiprogramalop
e3D8i3DB:  movlw d'61'
           goto fiprogramalop
e3DCi3DF:  movlw d'59'
           goto fiprogramalop
e3E0i3E3:  movlw d'57'
           goto fiprogramalop
e3E4i3E7:  movlw d'55'
           goto fiprogramalop
```



```

e3E8i3EB:  movlw d'53'
           goto  fiprogramalop
e3ECi3EF:  movlw d'51'
           goto  fiprogramalop
e3F0i3F3:  movlw d'49'
           goto  fiprogramalop
e3F4i3F7:  movlw d'47'
           goto  fiprogramalop
e3F8i3FB:  movlw d'45'
           goto  fiprogramalop
e3FCi3FF:  movlw d'43'
           goto  fiprogramalop

```

```

;*****
;***** Subrutinas aplicadas en el programa para el cálculo *****
;***** de la inferencia.*****
;*****

```

```

;*****
;***** Subrutina tantgeneral *****
;*****
;Esta rutina calcula el % de un valor. Este % está guardado en la
;posición de memoria GRPERT. El valor sobre el que se aplica el % está
;en OUT1 y el valor de salida de la función estará guardado en el
;registro W.

```

;EN EL DIAGRAMA DE FLUJO CORRESPONDE AL ESTADO (F)

tantgeneral:

```

      bcf   STATUS,C
      clrw
      rrf   OUT1,F           ;W=0
      btfsc GRPERT,4       ;OUT1 = OUT1/2
      addwf OUT1,W         ;miramos el bit 4, si es 1
                          ;W = W +OUT1
      bcf   STATUS,C       ;Si el bit 4 es 0, W = W
      rrf   OUT1,F         ;OUT1 = OUT1/2
      btfsc GRPERT,3       ;Si el bit 3 es igual a 1
      addwf OUT1,W         ;W = W +OUT1
      bcf   STATUS,C       ;Si el bit 3 es 0, W = W
      rrf   OUT1,F         ;OUT1 = OUT1/2
      btfsc GRPERT,2       ;Si el bit 2, es igual a 1
      addwf OUT1,W         ;W = W +OUT1
      bcf   STATUS,C       ;Si el bit 2 es 0, W = W
      rrf   OUT1,F         ;OUT1 = OUT1/2
      btfsc GRPERT,1       ;Si el bit 1 es igual 1
      addwf OUT1,W         ;W = W +OUT1
      rrf   OUT1,F         ;Si el bit 1 es 0, W = W
      bcf   STATUS,C       ;OUT1 = OUT1/2
      btfsc GRPERT,0       ;Si el bit 0 es igual a 1
      addwf OUT1,W         ;W = W +OUT1.
      goto fiprograma2op  ;El registro W contiene el valor
                          ;final con el % aplicado, por lo
                          ;tanto saltamos a la etiqueta
                          ;fiprograma2op donde se sumará el
                          ;valor del otro conjunto borroso
                          ;SALIDA_ANT2.

```

```

;*****
;***** Subrutina tantzero *****
;*****
;Esta rutina calcula el % de los K*CERROR de los dos PI's
;implementados, el % a aplicar está guardado en la posición de
;memoria GRPERT. El valor donde se ha de aplicar el % está en OUT1, y
;el resultado lo guardaremos en el registro W.

```

;EN EL DIAGRAMA DE FLUJO CORRESPONDE AL ESTADO ①

```

tantzero:  clrw          ;W=0
           bcf  STATUS,C
           rrf  OUT2,F    ;OUT2 = OUT2/2
           btfsc GRPERT,3 ;miramos el bit 3, si es = 1
           addwf OUT2,W   ;W = W +OUT2
           bcf  STATUS,C ;Si el bit 3 es 0, W = W
           rrf  OUT2,F    ;OUT2 = OUT2/2
           btfsc GRPERT,2 ;miramos el bit 2, si es = 1
           addwf OUT2,W   ;W = W +OUT2
           bcf  STATUS,C ;Si el bit 2 es 0, W = W
           rrf  OUT2,F    ;OUT2 = OUT2/2
           btfsc GRPERT,1 ;miramos el bit 1, si es = 1
           addwf OUT2,W   ;W = W +OUT2
           movwf SALIDA_ANT2 ;SALIDA_ANT2 = W, el valor
                               ;calculado.

           clrw          ;W=0
           bcf  STATUS,C
           rrf  OUT1,F    ;OUT1 = OUT1/2
           btfsc GRPERT,7 ;miramos el bit 7, si es = 1
           addwf OUT1,W   ;W = W +OUT1
           bcf  STATUS,C ;Si el bit 7 es 0, W = W
           rrf  OUT1,F    ;OUT1 = OUT1/2
           btfsc GRPERT,6 ;miramos el bit 6, si es = 1
           addwf OUT1,W   ;W = W +OUT1
           bcf  STATUS,C ;Si el bit 6 es 0, W = W
           rrf  OUT1,F    ;OUT1 = OUT1/2
           btfsc GRPERT,5 ;miramos el bit 5, si es = 1
           addwf OUT1,W   ;W = W +OUT1
           movwf SALIDA_ANT1 ;SALIDA_ANT1 = W, el valor

calculat.  addwf SALIDA_ANT2,F ;SALIDA_ANT2 = SALIDA_ANT2 +
                               ;SALIDA_ANT1, que será el incremto
                               ;de intensidad que se tendrá que
                               ;aplicar.

```

;Ya que antes de entrar en la función el valor ha sido multiplicado
;por 4, ahora realizaremos la operación inversa. De esta forma se ha
;calculado la salida de forma más precisa.

```

           bcf  STATUS,C
           rrf  SALIDA_ANT2,F ;/2
           bcf  STATUS,C
           rrf  SALIDA_ANT2,W ;/4, en el registro W está guardado
                               ;el incremento de la intensidad
                               ;que se ha de aplicar.

           btfsc SIGNECERROR,0 ;miramos el signo de la variable
           goto LAB            ; CERROR. Si es negativo, saltamos
           addwf PORTB_ANT,W   ; a LAB, sino sumamos a W, el valor
                               ; de la posición de memoria
                               ;PORT_ANT: De esta manera
                               ;W = Iref(anterior)+

```

```

; (K1*CERROR+K2*ERROR)*x%
; +K3*CERROR*y%.

btfsc STATUS,C
movlw 0xFF
goto fiprogramalop ;El resultado actualitzado está en
; el registro W y saltamos a
; la etiqueta correspondiente.

LAB: subwf PORTB_ANT,W ;restamos a W, el valor de la
; posición de memoria PORT_ANT. De
; esta forma W = Iref(anterior)+
; (-K1*CERROR+K2*ERROR)*x%
; -K3*CERROR*y%.

btfss STATUS,C
movlw d'11'
goto fiprogramalop ;El resultado actualitzado está en
; el registro W y saltamos a la
; etiqueta correspondiente.

;*****
;*****
;*****

fiprograma2op:
    addwf SALIDA_ANT2,W
    btfsc STATUS,C
    movlw 0xFF

;EN EL DIAGRAMA DE FLUJO CORRESPONDE AL ESTADO (G)

fiprogramalop: ;La salida está guardada en el
; registro W.
espera: btfss INTCON,T0IF ;Esperamos que se acabe el periodo
; de muestreo.
goto espera

;EN EL DIAGRAMA DE FLUJO CORRESPONDE AL ESTADO (J)

    movwf PORTB ;Cuando finaliza acabat el periodo
; muestreo se ha de actualizar la
; salida.
    goto principi ;Saltemos al inicio del programa,
; y esperamos un nuevo valor
; muestreado de l'error para ser
; evaluado.

END

```