```
;*********************************************************************
;                                                                    *
;    PROGRAMA QUE REALIZA LA ADQUISICIÓN DE DATOS, ELABORA EL        *
;     CONTROL DIFUSO Y MANEJA LA CANTIDAD DE PASOS QUE ENVÍA A LOS     *
;     MOTORES. EJECUTA EL TIRO Y REGRESA EL SISTEMA A SU CONDICIÓN     *
;     INICIAL                                                                                *
;                                                                   *
;*********************************************************************
;                                                                    *
;    FILENAME:        PROYECTO.ASM                                   *
;    DATE:          4-ABR-2003                                       *
;    FILE VERSION:  1.0                                              *
;                                                                    *
;    AUTHOR:        CRISTOBAL OCHOA LUNA-IRMING ISAAC HERNANDEZ BELLO *
;                                                                                  *
;    COMPANY:       F-302                                            *
;                                                                    *
;*********************************************************************
;                                                                    *
;    NOTES:         DISEÑADO PARA PIC16F874 A 20 MHz                 *
;                                                                    *
;*********************************************************************


      list      p=16f874            ; definir microprocesador
      #include <c-p16f874.inc>      ; definir variables del microprocesador

      __CONFIG _CP_OFF & _WDT_OFF & _BODEN_OFF & _PWRTE_OFF & _HS_OSC & _WRT_ENABLE_OFF & _LVP_OFF & _DEBUG_OFF &
_CPD_OFF


;*********************************************************************
;*                      DEFINICIÓN DE VARIABLES                      *
;*********************************************************************

; EN ESTA SECCIÓN SE DECLARAN LOS REGISTROS DE PROPÓSITO GENERAL QUE SE
; MANEJAN PARA EL FUNCIONAMIENTO DEL PROGRAMA

W_TEMP                      EQU 0X25
STATUS_TEMP             EQU   0X26
ULTIMO_PASO             EQU   0X27
ULTIMO_PASO_M1          EQU   0X28
ULTIMO_PASO_M2          EQU   0X29
CONT_PASOS_H            EQU   0X2A
CONT_PASOS_L            EQU 0X2B
VALOR_PRIMER_POT  EQU   0X2C
VALOR_SEGUNDO_POT EQU   0X2D
GAMMA                   EQU   0X2E
PASOS_MOTOR_1           EQU   0X2F
PASOS_MOTOR_2           EQU   0X30

IN1P1                   EQU 0X31
```

```
IN1P2                   EQU   0X32
IN1P3                   EQU   0X33
IN1P4                   EQU   0X34
IN1P5                   EQU   0X35
IN2P1                   EQU   0X36
IN2P2                   EQU 0X37
IN2P3                   EQU   0X38
IN2P4                   EQU   0X39
IN2P5                   EQU   0X3A

OUT1P1                        EQU 0X3B
OUT1P2                        EQU   0X3C
OUT1P3                  EQU   0X3D
OUT1P4                        EQU   0X3E
OUT1P5                        EQU   0X3F
OUT2P1                        EQU   0X40
OUT2P2                        EQU 0X41
OUT2P3                        EQU   0X42
OUT2P4                        EQU   0X43
OUT2P5                        EQU   0X44

MEM_IN1_C1              EQU   0X45
MEM_IN1_C2              EQU   0X46
MEM_IN1_C3              EQU   0X47
MEM_IN1_C4              EQU   0X48
MEM_IN1_C5              EQU   0X49

MEM_IN2_C1              EQU   0X4A
MEM_IN2_C2              EQU   0X4B
MEM_IN2_C3              EQU   0X4C
MEM_IN2_C4              EQU   0X4D
MEM_IN2_C5              EQU   0X4E

CLUSTERS_IN1            EQU   0X4F
CLUSTERS_IN2            EQU   0X50

MEMBRESIA               EQU 0X51
P1                            EQU 0X52
P2                            EQU 0X53
X                             EQU 0X54
NUM                           EQU   0X55
DEN                           EQU   0X56

CONTADOR                EQU   0X57

MULT_1_L                EQU   0X58
MULT_1_H                EQU   0X59
MULT_2                  EQU   0X5A
RES_MULT_L              EQU   0X5B
RES_MULT_H              EQU 0X5C
COMPARADOR_MULT         EQU   0X5D
```

```
DIVIDENDO_L             EQU   0X5E
DIVIDENDO_H             EQU   0X5F
DIVIDENDO_AUX           EQU   0X60
COCIENTE_L              EQU   0X61
COCIENTE_H              EQU 0X62
DIVISOR                       EQU   0X63

MEM_OUT_C1              EQU   0X64
MEM_OUT_C2              EQU   0X65
MEM_OUT_C3              EQU   0X66
MEM_OUT_C4              EQU   0X67
MEM_OUT_C5              EQU   0X68

SUBSTITUIR              EQU 0X69
MIN_1                   EQU   0X6A
MIN_2                   EQU   0X6B
INDICADOR_P1            EQU   0X6C
P1AUX                   EQU   0X6D
P2AUX                   EQU   0X6E
DEFUZZ_NUM_H            EQU   0X6F
DEFUZZ_NUM_L            EQU   0X70
DEFUZZ_DEN_H            EQU   0X71
DEFUZZ_DEN_L            EQU   0X72
MEM_1                   EQU   0X73
MEM_2                   EQU   0X74
MEM                           EQU   0X75
DEFUZZ_NUM_H_2          EQU   0X76

LIMITE1                       EQU   0X77
LIMITE2                       EQU   0X78
LIMITE3                       EQU   0X79
LIMITE4                       EQU   0X7A
LIMITE5                       EQU   0X7B
DIVISOR_TEMP            EQU   0X7C
RES_ALTURA              EQU   0X7D

RES_FUERZA              EQU   0X7E

;*****************************************************************
;*                        INICIO DE PROGRAMA                     *
;*****************************************************************

            ORG     0X000                       ; VECTOR DE INICIO DEL uCONTROLADOR
            CLRF    PCLATH                      ; COLOCARSE EN PÁGINA CORRECTA
            GOTO    CONFIG                      ; SALTAR AL INICIO DEL PROGRAMA

;*****************************************************************
;*                        VECTOR DE INTERRUPCIÓN                 *
;*****************************************************************

            ORG         0X004                   ; REUBICACIÓN DEL
            GOTO  ISR                           ; VECTOR DE INTERRUPCION
```

```
;*********************************************************************
;*                     INICIALIZACIÓN                                *
;*********************************************************************

; EN ESTA SECCIÓN SE DEFINEN LOS LÍMITES DE CADA CLUSTER, ESTA
; DIVISIÓN ES DE 5 ESPACIOS Y SE HACE PARA CADA VARIABLE DE ENTRADA
; “REACCIÓN” Y “DISTANCIA”, ASÍ COMO PARA LAS VARIABLES DE SALIDA
; “FUERZA” Y “ALTURA”, ESTO SE HACE ESCRIBIENDO EL VALOR DEL LÍMITE
; EN EL REGISTRO CORRESPONDIENTE. POR EJEMPLO DONDE IN1P1 ES EL PUNTO 1
; DE LA ENTRADA 1.

            ORG         0X10

CONFIG

            MOVLW 0X00
            MOVWF IN1P1
            MOVLW 0X3F
            MOVWF IN1P2
            MOVLW 0X7E
            MOVWF IN1P3
            MOVLW 0XBD
            MOVWF IN1P4
            MOVLW 0XFC
            MOVWF IN1P5

            MOVLW 0X00
            MOVWF IN2P1
            MOVLW 0X3F
            MOVWF IN2P2
            MOVLW 0X7E
            MOVWF IN2P3
            MOVLW 0XBD
            MOVWF IN2P4
            MOVLW 0XFC
            MOVWF IN2P5

            MOVLW 0X00
            MOVWF OUT1P1
            MOVLW 0X3F
            MOVWF OUT1P2
            MOVLW 0X7E
            MOVWF OUT1P3
            MOVLW 0XBD
            MOVWF OUT1P4
            MOVLW 0XFC
            MOVWF OUT1P5

            MOVLW 0X00
            MOVWF OUT2P1
            MOVLW 0X3F
```

```
            MOVWF OUT2P2
            MOVLW 0X7E
            MOVWF OUT2P3
            MOVLW 0XBD
            MOVWF OUT2P4
            MOVLW 0XFC
            MOVWF OUT2P5

;CONFIGURACIÓN DEL CONVERTIDOR A/D
;*********************************

            BSF         STATUS,RP0

            MOVLW 0X04                  ; RESULTADO JUSTIFICADO A LA IZQUIERDA,
            MOVWF ADCON1                      ; CONFIGURAR AN0, AN1 Y AN3 COMO
                                                    ; ENTRADAS ANALÓGICAS Y LOS DEMÁS PINES
            MOVLW 0X0B                  ; DEL PUERTO A, COMO I/O DIGITALES, Y
            MOVWF TRISA                 ; SU VALOR CORRESPONDIENTE EN EL TRIS

            BCF         STATUS,RP0


            MOVLW 0X80                  ; RELOJ DE CONVERSIÓN = Fosc/32, CANAL
            MOVWF ADCON0                      ; SELECCIONADO AN0, MODULO APAGADO

;CONFIGURACIÓN DEL TIMER0
;************************

            BSF         STATUS,RP0

            MOVLW 0X00                  ; RELOJ INTERNO, PREESCALA 1:2 (TMR0)
            MOVWF OPTION_REG

;CONFIGURACIÓN DEL TIMER1
;************************

            BCF         STATUS,RP0
            MOVLW 0X0E                  ; OSCILADOR HABILITADO, RELOJ EXTERNO
            MOVWF T1CON                 ; NO SINCRONIZADO, TIMER APAGADO
            BSF         STATUS,RP0

;CONFIGURACIÓN DEL PUERTO B
;**************************

            MOVLW 0X80                  ; PIN7 ENTRADA, LOS DEMÁS SON SALIDAS
            MOVWF TRISB

            ; PULL-UPS ACTIVADAS EN FLANCO DE BAJADA CON EL 05
            ; CARGADO EN EL OPTION REGISTER EN LA CONFIGURACIÓN
            ; DEL TIMER 0

;CONFIGURACIÓN DEL PUERTO C
```

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

            MOVLW 0X00                  ; PINES DEL PUERTO C, COMO SALIDAS
            MOVWF TRISC

;CONFIGURACION DEL PUERTO D
;*************************

            MOVLW 0X00                  ; PINES DEL PUERTO D, COMO SALIDAS
            MOVWF TRISD

;CONFIGURAR DIRECCIONAMIENTO INDIRECTO
;***********************************

            MOVLW PORTB                 ; SE APUNTA AL PUERTO DEL PRIMER MOTOR
            MOVWF FSR                         ; DIRECCIONAMIENTO INDIRECTO PARA REUTILIZAR CÓDIGO

;*********************************************************************
;*                   INICIALIZACIÓN                                  *
;*********************************************************************

; EN ESTA SECCIÓN SE LIMPIAN LOS PUERTOS Y SE MUEVE EL PRIMER PASO
; A AMBOS MOTORES PARA COMENZAR A GIRAR A PARTIR DE DICHO PASO
; CUANDO SE LE INIDÍQUE MOTOR 1 Ó 2.
; ASÍ COMO TAMBIÉN SE INICIALIZAN LOS CONTADORES DE PASOS QUE DAN LA
; PREESCALA Y LOS PASOS QUE SE MOVERÁ EL MOTOR.

            BCF         STATUS,RP0

            CLRF  PORTA
            CLRF  PORTB
            CLRF  PORTC
            CLRF  PORTD
            CLRF  ALFA
            CLRF  BETA

            MOVLW 0X0A
            MOVWF PORTB
            MOVWF ULTIMO_PASO_M1
            MOVWF ULTIMO_PASO

            MOVLW 0X0A
            MOVWF PORTD
            MOVWF ULTIMO_PASO_M2

            MOVLW 0X01
            MOVWF CONT_PASOS_H
            MOVLW 0X0B
            MOVWF CONT_PASOS_L

;INICIALIZACIÓN DE INTERRUPCIONES
;*******************************
```

```
            BCF         INTCON,RBIF             ; LIMPIA BANDERA DE INTERRUPCIÓN RB
            BSF         INTCON,RBIE             ; HABILITA INTERRUP. DE PULL-UP, LEE
            MOVF  PORTB,0                       ; PUERTO B, PARA DETECTAR EL CAMBIO.
            BSF         INTCON,GIE              ; HABILITAR INTERRUPCIONES GENERALES
            BSF         INTCON,PEIE             ; HABILITAR INTERRUPCIONES DE PERIFERICOS


            BCF         STATUS,RP0
            GOTO  MAIN


;*****************************************************************
;*                          PROGRAMA PRINCIPAL                   *
;*****************************************************************

            ORG         0X080

MAIN

; SISTEMA QUE CONTROLA EL FLUJO DEL PROGRAMA; MEDIANTE CIERTOS INDICADORES,
; INDICA QUE SECCIÓN DEL PROGRAMA DEBE REALIZAR SU PROCEDIMIENTO. POR
; EJEMPLO: LA DIRECCIÓN EN QUE GIRARÁN LOS MOTORES, EL CONTROL DIFUSO O LA
; EJECUCIÓN DEL TIRO.

            BTFSC BETA,SYSTEM_OK
            CALL  CICLO_LISTO

            BTFSC ALFA,MOTORES
            GOTO  DIRECCION

            BTFSC BETA,TIRO
            CALL  EJECUTAR_TIRO

            BTFSC ALFA,FUZZY
            CALL  FUZZY_LOGIC

            NOP
            GOTO  MAIN


;-------------------------------------------------------------------------

;********
DIRECCION
;********

; EN ESTA SECCIÓN SE DETERMINA EL SENTIDO EN EL QUE GIRARÁN LOS MOTORES.

            BTFSC ALFA,MOTOR2
            CALL  SENTIDO_MOTOR2
            BTFSS ALFA,MOTOR2
```

```
            CALL  SENTIDO_MOTOR1
            BSF         ALFA,PRIMER_PASO
            BCF         ALFA,MOTORES
            BTFSS ALFA,SENTIDO
            CALL  DERECHA
            BTFSC ALFA,SENTIDO
            CALL  IZQUIERDA
            BSF         INTCON,PEIE
            BSF         INTCON,GIE
            GOTO  MAIN

SENTIDO_MOTOR1
;*************

; EN ESTA SECCIÓN SE ANALIZAN LOS INDICADORES DEL SENTIDO QUE TENDRA EL MOTOR 1,
; PARA QUE EN LA SECCIÓN DE DIRECCIÓN PUEDA INDICARSE HACÍA DONDE GIRAR.

            BTFSC BETA,SENTIDO_1
            BSF         ALFA,SENTIDO
            BTFSS BETA,SENTIDO_1
            BCF         ALFA,SENTIDO
            RETURN

SENTIDO_MOTOR2
;*************

; EN ESTA SECCIÓN SE ANALIZAN LOS INDICADORES DEL SENTIDO QUE TENDRA EL MOTOR 2,
; PARA QUE EN LA SECCIÓN DE DIRECCIÓN PUEDA INDICARSE HACÍA DONDE GIRAR.

            BTFSC BETA,SENTIDO_2
            BSF         ALFA,SENTIDO
            BTFSS BETA,SENTIDO_2
            BCF         ALFA,SENTIDO
            RETURN

;-------------------------------------------------------------------------------

;-------------------------------------------------------------------------------

;************
EJECUTAR_TIRO
;************

; EN ESTA PARTE SE LE INDICA AL SISTEMA QUE REALICE EL TIRO, MEDIANTE UN PULSO
; Y SE MANTIENE POR UN TIEMPO DE 1s PARA DAR EL TIEMPO SUFICIENTE A LA EJECUCIÓN DEL TIRO.

            BCF         INTCON,GIE
            BCF         INTCON,PEIE

            BSF         PORTA,5
            BCF         BETA,TIRO
```

```
            MOVLW 0X00
            MOVWF TMR1L
            MOVLW 0X80
            MOVWF TMR1H
            BCF         PIR1,TMR1IF
            BSF         STATUS,RP0
            BSF         PIE1,TMR1IE
            BCF         STATUS,RP0

            BSF         T1CON,TMR1ON

            BCF         INTCON,T0IE

            BSF         INTCON,GIE
            BSF         INTCON,PEIE

            RETURN

;-------------------------------------------------------------------------------

;-------------------------------------------------------------------------------

;**********
CICLO_LISTO
;**********

; INDICA QUE EL SISTEMA SE HA REESTABLECIDO Y REINICIA

            BCF         INTCON,GIE

            CLRF  ALFA
            CLRF  BETA

            BCF         INTCON,RBIF
            MOVF  PORTB,0
            BSF         INTCON,RBIE

            BSF         INTCON,GIE

            RETURN

;-------------------------------------------------------------------------------

;-------------------------------------------------------------------------------

;**********
FUZZY_LOGIC
;**********

; SECCIÓN QUE CONTROLA LA OPERACIÓN DEL CONTROLADOR DIFUSO
; MEDIANTE LAS SUBRUTINAS DESCRITAS A CONTINUACIÓN, Y REALIZA
; TAMBIÉN AL FINAL LA PARTE DE AJUSTE DE LOS RESULTADOS A LOS
```

```
; PASOS Y DIRECCIÓN QUE SE DEBEN MOVER LOS MOTORES.

            BCF         INTCON,GIE
            BCF         INTCON,PEIE

            CLRF  MEM_IN1_C1
            CLRF  MEM_IN1_C2
            CLRF  MEM_IN1_C3
            CLRF  MEM_IN1_C4
            CLRF  MEM_IN1_C5
            CLRF  MEM_IN2_C1
            CLRF  MEM_IN2_C2
            CLRF  MEM_IN2_C3
            CLRF  MEM_IN2_C4
            CLRF  MEM_IN2_C5
            CLRF  MEM_OUT_C1
            CLRF  MEM_OUT_C2
            CLRF  MEM_OUT_C3
            CLRF  MEM_OUT_C4
            CLRF  MEM_OUT_C5

            CALL  CALCULAR_MEMBRESIAS_IN1
            CALL  CALCULAR_MEMBRESIAS_IN2

            CALL  INFERENCIA_IN1
            CALL  DEFUZIFICAR1

            CLRF  MEM_OUT_C1
            CLRF  MEM_OUT_C2
            CLRF  MEM_OUT_C3
            CLRF  MEM_OUT_C4
            CLRF  MEM_OUT_C5

            CALL  INFERENCIA_IN2
            CALL  DEFUZIFICAR2

; SECCIÓN QUE AJUSTA LOS RESULTADOS DEL CONTROL DIFUSO A MOVIMIENTO DE MOTORES

            MOVLW 0X7E
            SUBWF RES_ALTURA,0
            MOVWF PASOS_MOTOR_2
            BTFSC STATUS,C
            BCF         BETA,SENTIDO_2
            BTFSS STATUS,C
            CALL  RES_NEG_ALTURA

            MOVLW 0X7E
            SUBWF RES_FUERZA,0
            MOVWF PASOS_MOTOR_1
            MOVWF CONT_PASOS_H
            BTFSC STATUS,C
            BCF         BETA,SENTIDO_1
```

```
            BTFSS STATUS,C
            CALL  RES_NEG_FUERZA

            MOVLW PORTB
            MOVWF FSR

            BCF         ALFA,FUZZY
            BSF         ALFA,MOTORES

            RETURN

RES_NEG_FUERZA
;*************

            MOVF  RES_FUERZA,0
            SUBLW 0X7E
            MOVWF PASOS_MOTOR_1
            MOVWF CONT_PASOS_H
            BSF         BETA,SENTIDO_1
            RETURN

RES_NEG_ALTURA
;*************

            MOVF  RES_ALTURA,0
            SUBLW 0X7E
            MOVWF PASOS_MOTOR_2
            BSF         BETA,SENTIDO_2
            RETURN

;-------------------------------------------------------------------------
CALCULAR_MEMBRESIAS_IN1

; EN ESTA SECCIÓN SE CALCULA LA MEMBRESÍA DEL VALOR CORRESPONDIENTE AL PRIMER ESPACIO DE ENTRADA.
; ESTO SE HACE MEDIANTE COMPARACIONES; ES DECIR, MEDIANTE RESTAS ENTRE EL DATO INGRESADO Y LOS
; LÍMITES DE LOS CLUSTERS SE ANALIZA SI EL VALOR DE ENTRADA PERTENECE AL PRIMER CLUSTER, SI NO,
; REALIZA EL MISMO PROCEDIMIENTO CON LOS LÍMITES DEL SEGUNDO CLUSTER Y ASÍ CONSECUTIVAMENTE. ENVÍA
; A LA SUBRUTINA CORREPONDIENTE A ENCONTRAR EL VALOR DE LA MEMBRESÍA DE LOS CLUSTERS INDICADOS.

            BSF         FUZZY_REG,IN1
            BCF         FUZZY_REG,IN2

            MOVF  VALOR_PRIMER_POT,0
            MOVWF X
            BSF         FUZZY_REG,PENDIENTE_POS

;CHECAR_CLUSTER1_2

            MOVF  VALOR_PRIMER_POT,0
            SUBWF IN1P2,0

            BTFSS STATUS,C
```

```
            GOTO  CHECAR_CLUSTER2_3
            MOVLW 0X03
            MOVWF CLUSTERS_IN1
            GOTO  IN1_CLUSTER1_2

CHECAR_CLUSTER2_3

            MOVF  VALOR_PRIMER_POT,0
            SUBWF IN1P3,0

            BTFSS STATUS,C
            GOTO  CHECAR_CLUSTER3_4
            MOVLW 0X06
            MOVWF CLUSTERS_IN1
            GOTO  IN1_CLUSTER2_3

CHECAR_CLUSTER3_4

            MOVF  VALOR_PRIMER_POT,0
            SUBWF IN1P4,0

            BTFSS STATUS,C
            GOTO  IN1_CLUSTER4_5
            MOVLW 0X0C
            MOVWF CLUSTERS_IN1
            GOTO  IN1_CLUSTER3_4

; EN ESTA SECCIÓN, UNA VEZ QUE SE TIENE EL CLUSTER AL QUE PERTENECE EL DATO DE ENTRADA,
; SE PROPORCIONAN LOS LÍMITES DEL CLUSTER Y SE PROSIGUE CON LA RUTINA QUE CALCULA LA
; PERTENENCIA DE DICHO VALOR EN EL ESPACIO DE ENTRADA.

IN1_CLUSTER1_2

            MOVF  IN1P1,0
            MOVWF P1
            MOVF  IN1P2,0
            MOVWF P2

            CALL  SUBRUTINA_CALC_MEM
            MOVF  MEMBRESIA,0
            MOVWF MEM_IN1_C1

            SUBLW 0X64
            MOVWF MEM_IN1_C2

            RETURN

IN1_CLUSTER2_3

            MOVF  IN1P2,0
            MOVWF P1
            MOVF  IN1P3,0
```

```
            MOVWF P2

            CALL  SUBRUTINA_CALC_MEM
            MOVF  MEMBRESIA,0
            MOVWF MEM_IN1_C2

            SUBLW 0X64
            MOVWF MEM_IN1_C3

            RETURN

IN1_CLUSTER3_4

            MOVF  IN1P3,0
            MOVWF P1
            MOVF  IN1P4,0
            MOVWF P2

            CALL  SUBRUTINA_CALC_MEM
            MOVF  MEMBRESIA,0
            MOVWF MEM_IN1_C3

            SUBLW 0X64
            MOVWF MEM_IN1_C4

            RETURN

IN1_CLUSTER4_5

            MOVLW 0X18
            MOVWF CLUSTERS_IN1

            MOVF  IN1P4,0
            MOVWF P1
            MOVF  IN1P5,0
            MOVWF P2


            CALL  SUBRUTINA_CALC_MEM
            MOVF  MEMBRESIA,0
            MOVWF MEM_IN1_C4

            SUBLW 0X64
            MOVWF MEM_IN1_C5

            RETURN

;-------------------------------------------------------------------------

; EN ESTA SECCIÓN SE CALCULA LA MEMBRESÍA DEL VALOR CORRESPONDIENTE AL SEGUNDO ESPACIO DE ENTRADA.
; ESTO SE HACE MEDIANTE COMPARACIONES; ES DECIR, MEDIANTE RESTAS ENTRE EL DATO INGRESADO Y LOS
; LÍMITES DE LOS CLUSTERS SE ANALIZA SI EL VALOR DE ENTRADA PERTENECE AL PRIMER CLUSTER, SI NO,
```

```
; REALIZA EL MISMO PROCEDIMIENTO CON LOS LÍMITES DEL SEGUNDO CLUSTER Y ASÍ CONSECUTIVAMENTE. ENVÍA
; A LA SUBRUTINA CORRESPONDIENTE A ENCONTRAR EL VALOR DE LA MEMBRESÍA DE LOS CLUSTERS INDICADOS.

;*********************************************
CALCULAR_MEMBRESIAS_IN2


            BCF         FUZZY_REG,IN1
            BSF         FUZZY_REG,IN2

            MOVF  VALOR_SEGUNDO_POT,0
            MOVWF X
            BSF         FUZZY_REG,PENDIENTE_POS

;CHECAR_2_CLUSTER1_2

            MOVF  VALOR_SEGUNDO_POT,0
            SUBWF IN2P2,0

            BTFSS STATUS,C
            GOTO  CHECAR_2_CLUSTER2_3
            MOVLW 0X03
            MOVWF CLUSTERS_IN2
            GOTO  IN2_CLUSTER1_2

CHECAR_2_CLUSTER2_3

            MOVF  VALOR_SEGUNDO_POT,0
            SUBWF IN2P3,0

            BTFSS STATUS,C
            GOTO  CHECAR_2_CLUSTER3_4
            MOVLW 0X06
            MOVWF CLUSTERS_IN2
            GOTO  IN2_CLUSTER2_3

CHECAR_2_CLUSTER3_4

            MOVF  VALOR_SEGUNDO_POT,0
            SUBWF IN2P4,0

            BTFSS STATUS,C
            GOTO  IN2_CLUSTER4_5
            MOVLW 0X0C
            MOVWF CLUSTERS_IN2
            GOTO  IN2_CLUSTER3_4

; EN ESTA SECCIÓN, UNA VEZ QUE SE TIENE EL CLUSTER AL QUE PERTENECE EL DATO DE ENTRADA,
; SE PROPORCIONAN LOS LÍMITES DEL CLUSTER Y SE PROSIGUE CON LA RUTINA QUE CALCULA LA
; PERTENENCIA DE DICHO VALOR EN EL ESPACIO DE ENTRADA.

IN2_CLUSTER1_2
```

```
            MOVF  IN2P1,0
            MOVWF P1
            MOVF  IN2P2,0
            MOVWF P2

            CALL  SUBRUTINA_CALC_MEM
            MOVF  MEMBRESIA,0
            MOVWF MEM_IN2_C1

            SUBLW 0X64
            MOVWF MEM_IN2_C2

            RETURN

IN2_CLUSTER2_3

            MOVF  IN2P2,0
            MOVWF P1
            MOVF  IN2P3,0
            MOVWF P2

            CALL  SUBRUTINA_CALC_MEM
            MOVF  MEMBRESIA,0
            MOVWF MEM_IN2_C2

            SUBLW 0X64
            MOVWF MEM_IN2_C3

            RETURN

IN2_CLUSTER3_4

            MOVF  IN2P3,0
            MOVWF P1
            MOVF  IN2P4,0
            MOVWF P2

            CALL  SUBRUTINA_CALC_MEM
            MOVF  MEMBRESIA,0
            MOVWF MEM_IN2_C3

            SUBLW 0X64
            MOVWF MEM_IN2_C4

            RETURN

IN2_CLUSTER4_5

            MOVLW 0X18
            MOVWF CLUSTERS_IN2
```

```
            MOVF  IN2P4,0
            MOVWF P1
            MOVF  IN2P5,0
            MOVWF P2

            CALL  SUBRUTINA_CALC_MEM
            MOVF  MEMBRESIA,0
            MOVWF MEM_IN2_C4

            SUBLW 0X64
            MOVWF MEM_IN2_C5

            RETURN

;-------------------------------------------------------------------------------

;-------------------------------------------------------------------------------

; EN ESTA PARTE DEL PROGRAMA, SE DETERMINAN LAS REGLAS A APLICARSE EN EL ESPACIO 1.
; ESTO SE REALIZA, MEDIANTE COMPARACIONES, SI SE CUMPLEN LOS ANTECEDENTES (BITS EN 1
; DETERMINADOS POR LAS MEMBRESÍAS) SE TOMA EL CONSECUENTE.

INFERENCIA_IN1

            BTFSC CLUSTERS_IN1,0
            CALL  INF_1_SUBRUT_1

            BTFSC CLUSTERS_IN1,1
            CALL  INF_1_SUBRUT_2

            BTFSC CLUSTERS_IN1,2
            CALL  INF_1_SUBRUT_3

            BTFSC CLUSTERS_IN1,3
            CALL  INF_1_SUBRUT_4

            BTFSC CLUSTERS_IN1,4
            CALL  INF_1_SUBRUT_5

            RETURN

INF_1_SUBRUT_1

            MOVF  MEM_IN1_C1,0
            MOVWF MEM_OUT_C1

            RETURN

INF_1_SUBRUT_2

            MOVF  MEM_IN1_C2,0
            MOVWF MEM_OUT_C2
```

```
            RETURN

INF_1_SUBRUT_3

            MOVF  MEM_IN1_C3,0
            MOVWF MEM_OUT_C3

            RETURN

INF_1_SUBRUT_4

            MOVF  MEM_IN1_C4,0
            MOVWF MEM_OUT_C4

            RETURN

INF_1_SUBRUT_5

            MOVF  MEM_IN1_C5,0
            MOVWF MEM_OUT_C5

            RETURN

;-------------------------------------------------------------------------------

;-------------------------------------------------------------------------------

; SE REALIZA LA MISMA APLICACIÓN DE REGLAS, PERO PARA EL SEGUNDO ESPACIO. AL SER
; MAYOR CANTIDAD DE REGLAS SE TIENE MAYOR NÚMERO DE COMPARACIONES, PERO EL
; FUNCIONAMIENTO BÁSICAMENTE ES EL MISMO.

INFERENCIA_IN2

            BTFSC CLUSTERS_IN2,4
            CALL  INF_2_SUBRUT_1

            BTFSC CLUSTERS_IN2,3
            CALL  INF_2_SUBRUT_2

            BTFSC CLUSTERS_IN2,2
            CALL  INF_2_SUBRUT_3

            BTFSC CLUSTERS_IN2,1
            CALL  INF_2_SUBRUT_4

            BTFSC CLUSTERS_IN2,0
            CALL  INF_2_SUBRUT_5

            RETURN

INF_2_SUBRUT_1
```

```
; SUBRUTINA: SI DISTANCIA ES MUY_LARGO ENTONCES FUERZA ES MUY_FUERTE

            MOVF  MEM_IN2_C5,0
            MOVWF SUBSTITUIR
            MOVLW MEM_OUT_C5
            MOVWF FSR
            CALL  SUBRUTINA_SUSTITUCION

            RETURN

INF_2_SUBRUT_2

; SUBRUTINA QUE CHECA LAS REGLAS QUE TENGAN DISTANCIA LARGA

            MOVF  MEM_IN2_C4,0
            MOVWF MIN_1

            BTFSC CLUSTERS_IN1,4
            CALL  INF_2_SUBRUT_2_1

            BTFSC CLUSTERS_IN1,3
            CALL  INF_2_SUBRUT_2_2

            BTFSC CLUSTERS_IN1,2
            CALL  INF_2_SUBRUT_2_3

            BTFSC CLUSTERS_IN1,1
            CALL  INF_2_SUBRUT_2_4

            BTFSC CLUSTERS_IN1,0
            CALL  INF_2_SUBRUT_2_5

            RETURN

INF_2_SUBRUT_2_1

; SUBRUTINA: SI DISTANCIA ES LARGO Y REACCION ES AVANZA_MUCHO

            MOVF  MEM_IN1_C5,0
            MOVWF MIN_2
            CALL  SUBRUTINA_MINIMO

            MOVLW MEM_OUT_C4
            MOVWF FSR
            CALL  SUBRUTINA_SUSTITUCION

            RETURN

INF_2_SUBRUT_2_2

; SUBRUTINA: SI DISTANCIA ES LARGO Y REACCION ES AVANZA_POCO
```

```
            MOVF  MEM_IN1_C4,0
            MOVWF MIN_2
            CALL  SUBRUTINA_MINIMO

            MOVLW MEM_OUT_C4
            MOVWF FSR
            CALL  SUBRUTINA_SUSTITUCION

            RETURN

INF_2_SUBRUT_2_3

; SUBRUTINA: SI DISTANCIA ES LARGO Y REACCION ES DETENERSE

            MOVF  MEM_IN1_C3,0
            MOVWF MIN_2
            CALL  SUBRUTINA_MINIMO

            MOVLW MEM_OUT_C4
            MOVWF FSR
            CALL  SUBRUTINA_SUSTITUCION

            RETURN

INF_2_SUBRUT_2_4

; SUBRUTINA: SI DISTANCIA ES LARGO Y REACCION ES REGRESA_POCO

            MOVF  MEM_IN1_C2,0
            MOVWF MIN_2
            CALL  SUBRUTINA_MINIMO

            MOVLW MEM_OUT_C5
            MOVWF FSR
            CALL  SUBRUTINA_SUSTITUCION

            RETURN

INF_2_SUBRUT_2_5

; SUBRUTINA: SI DISTANCIA ES LARGO Y REACCION ES REGRESA_MUCHO

            MOVF  MEM_IN1_C1,0
            MOVWF MIN_2
            CALL  SUBRUTINA_MINIMO

            MOVLW MEM_OUT_C5
            MOVWF FSR
            CALL  SUBRUTINA_SUSTITUCION

            RETURN
```

```
;*****

INF_2_SUBRUT_3

; SUBRUTINA QUE CHECA LAS REGLAS QUE TENGAN DISTANCIA MEDIA

            MOVF  MEM_IN2_C3,0
            MOVWF MIN_1

            BTFSC CLUSTERS_IN1,4
            CALL  INF_2_SUBRUT_3_1

            BTFSC CLUSTERS_IN1,3
            CALL  INF_2_SUBRUT_3_2

            BTFSC CLUSTERS_IN1,2
            CALL  INF_2_SUBRUT_3_3


            BTFSC CLUSTERS_IN1,1
            CALL  INF_2_SUBRUT_3_4

            BTFSC CLUSTERS_IN1,0
            CALL  INF_2_SUBRUT_3_5

            RETURN

INF_2_SUBRUT_3_1

; SUBRUTINA: SI DISTANCIA ES MEDIO Y REACCION ES AVANZA_MUCHO

            MOVF  MEM_IN1_C5,0
            MOVWF MIN_2
            CALL  SUBRUTINA_MINIMO

            MOVLW MEM_OUT_C3
            MOVWF FSR
            CALL  SUBRUTINA_SUSTITUCION

            RETURN

INF_2_SUBRUT_3_2

; SUBRUTINA: SI DISTANCIA ES MEDIO Y REACCION ES AVANZA_POCO

            MOVF  MEM_IN1_C4,0
            MOVWF MIN_2
            CALL  SUBRUTINA_MINIMO

            MOVLW MEM_OUT_C3
            MOVWF FSR
```

```
            CALL  SUBRUTINA_SUSTITUCION

            RETURN

INF_2_SUBRUT_3_3

; SUBRUTINA: SI DISTANCIA ES MEDIO Y REACCION ES DETENERSE

            MOVF  MEM_IN1_C3,0
            MOVWF MIN_2
            CALL  SUBRUTINA_MINIMO

            MOVLW MEM_OUT_C3
            MOVWF FSR
            CALL  SUBRUTINA_SUSTITUCION

            RETURN

INF_2_SUBRUT_3_4

; SUBRUTINA: SI DISTANCIA ES MEDIO Y REACCION ES REGRESA_POCO

            MOVF  MEM_IN1_C2,0
            MOVWF MIN_2
            CALL  SUBRUTINA_MINIMO

            MOVLW MEM_OUT_C5
            MOVWF FSR
            CALL  SUBRUTINA_SUSTITUCION

            RETURN

INF_2_SUBRUT_3_5

; SUBRUTINA: SI DISTANCIA ES MEDIO Y REACCION ES REGRESA_MUCHO

            MOVF  MEM_IN1_C1,0
            MOVWF MIN_2
            CALL  SUBRUTINA_MINIMO

            MOVLW MEM_OUT_C5
            MOVWF FSR
            CALL  SUBRUTINA_SUSTITUCION

            RETURN

;*****

INF_2_SUBRUT_4

; SUBRUTINA QUE CHECA LAS REGLAS QUE TENGAN DISTANCIA CORTA
```

```
            MOVF  MEM_IN2_C2,0
            MOVWF MIN_1

            BTFSC CLUSTERS_IN1,4
            CALL  INF_2_SUBRUT_4_1

            BTFSC CLUSTERS_IN1,3
            CALL  INF_2_SUBRUT_4_2

            BTFSC CLUSTERS_IN1,2
            CALL  INF_2_SUBRUT_4_3

            BTFSC CLUSTERS_IN1,1
            CALL  INF_2_SUBRUT_4_4

            BTFSC CLUSTERS_IN1,0
            CALL  INF_2_SUBRUT_4_5

            RETURN

INF_2_SUBRUT_4_1

; SUBRUTINA: SI DISTANCIA ES CORTO Y REACCION ES AVANZA_MUCHO

            MOVF  MEM_IN1_C5,0
            MOVWF MIN_2
            CALL  SUBRUTINA_MINIMO

            MOVLW MEM_OUT_C3
            MOVWF FSR
            CALL  SUBRUTINA_SUSTITUCION

            RETURN

INF_2_SUBRUT_4_2

; SUBRUTINA: SI DISTANCIA ES CORTO Y REACCION ES AVANZA_POCO

            MOVF  MEM_IN1_C4,0
            MOVWF MIN_2
            CALL  SUBRUTINA_MINIMO

            MOVLW MEM_OUT_C3
            MOVWF FSR
            CALL  SUBRUTINA_SUSTITUCION

            RETURN

INF_2_SUBRUT_4_3

; SUBRUTINA: SI DISTANCIA ES CORTO Y REACCION ES DETENERSE
```

```
            MOVF  MEM_IN1_C3,0
            MOVWF MIN_2
            CALL  SUBRUTINA_MINIMO

            MOVLW MEM_OUT_C2
            MOVWF FSR
            CALL  SUBRUTINA_SUSTITUCION

            RETURN

INF_2_SUBRUT_4_4

; SUBRUTINA: SI DISTANCIA ES CORTO Y REACCION ES REGRESA_POCO

            MOVF  MEM_IN1_C2,0
            MOVWF MIN_2
            CALL  SUBRUTINA_MINIMO

            MOVLW MEM_OUT_C4
            MOVWF FSR
            CALL  SUBRUTINA_SUSTITUCION

            RETURN

INF_2_SUBRUT_4_5

; SUBRUTINA: SI DISTANCIA ES CORTO Y REACCION ES REGRESA_MUCHO

            MOVF  MEM_IN1_C1,0
            MOVWF MIN_2
            CALL  SUBRUTINA_MINIMO

            MOVLW MEM_OUT_C5
            MOVWF FSR
            CALL  SUBRUTINA_SUSTITUCION

            RETURN

;*****

INF_2_SUBRUT_5

; SUBRUTINA QUE CHECA LAS REGLAS QUE TENGAN DISTANCIA MUY_CORTA

            MOVF  MEM_IN2_C1,0
            MOVWF MIN_1

            BTFSC CLUSTERS_IN1,4
            CALL  INF_2_SUBRUT_5_1

            BTFSC CLUSTERS_IN1,3
            CALL  INF_2_SUBRUT_5_2
```

```
            BTFSC CLUSTERS_IN1,2
            CALL  INF_2_SUBRUT_5_3

            BTFSC CLUSTERS_IN1,1
            CALL  INF_2_SUBRUT_5_4

            BTFSC CLUSTERS_IN1,0
            CALL  INF_2_SUBRUT_5_5

            RETURN

INF_2_SUBRUT_5_1

; SUBRUTINA: SI DISTANCIA ES MUY_CORTO Y REACCION ES AVANZA_MUCHO

            MOVF  MEM_IN1_C5,0
            MOVWF MIN_2
            CALL  SUBRUTINA_MINIMO

            MOVLW MEM_OUT_C2

            MOVWF FSR
            CALL  SUBRUTINA_SUSTITUCION

            RETURN

INF_2_SUBRUT_5_2

; SUBRUTINA: SI DISTANCIA ES MUY_CORTO Y REACCION ES AVANZA_POCO

            MOVF  MEM_IN1_C4,0
            MOVWF MIN_2
            CALL  SUBRUTINA_MINIMO

            MOVLW MEM_OUT_C2
            MOVWF FSR
            CALL  SUBRUTINA_SUSTITUCION

            RETURN

INF_2_SUBRUT_5_3

; SUBRUTINA: SI DISTANCIA ES MUY_CORTO Y REACCION ES DETENERSE

            MOVF  MEM_IN1_C3,0
            MOVWF MIN_2
            CALL  SUBRUTINA_MINIMO

            MOVLW MEM_OUT_C2
            MOVWF FSR
            CALL  SUBRUTINA_SUSTITUCION
```

```
            RETURN

INF_2_SUBRUT_5_4

; SUBRUTINA: SI DISTANCIA ES MUY_CORTO Y REACCION ES REGRESA_POCO

            MOVF  MEM_IN1_C2,0
            MOVWF MIN_2
            CALL  SUBRUTINA_MINIMO

            MOVLW MEM_OUT_C4
            MOVWF FSR
            CALL  SUBRUTINA_SUSTITUCION

            RETURN

INF_2_SUBRUT_5_5

; SUBRUTINA: SI DISTANCIA ES MUY_CORTO Y REACCION ES REGRESA_MUCHO

            MOVF  MEM_IN1_C1,0
            MOVWF MIN_2
            CALL  SUBRUTINA_MINIMO

            MOVLW MEM_OUT_C4
            MOVWF FSR
            CALL  SUBRUTINA_SUSTITUCION

            RETURN

;-------------------------------------------------------------------------

;-------------------------------------------------------------------------

; EN ESTA SECCIÓN SE DEFUZIFICAN LAS SALIDAS, SE EVALÚA CADA SEGMENTO DEL
; ESPACIO DE SALIDA CORRESPONDIENTE Y SE DETERMINA SU MEMBRESÍA SI EXISTE
; SE VA EJECUTANDO LA ECUACIÓN (SUMATORIA) DEL MÉTODO DEL CENTROIDE DE LAS
; MEMBRESÍAS, PARA FINALMENTE REALIZAR LA DIVISIÓN Y TENER EL CENTRO DE
; GRAVEDAD DEL POLÍGONO DE SALIDA.

DEFUZIFICAR1

            MOVF  OUT1P1,0
            MOVWF LIMITE1
            MOVF  OUT1P2,0
            MOVWF LIMITE2
            MOVF  OUT1P3,0
            MOVWF LIMITE3
            MOVF  OUT1P4,0
            MOVWF LIMITE4
            MOVF  OUT1P5,0
```

```
            MOVWF LIMITE5

            CALL  DEFUZIFICAR
            CALL  CENTROIDE
            MOVF  COCIENTE_L,0
            MOVWF RES_ALTURA

            RETURN


CENTROIDE

            MOVF  DEFUZZ_DEN_H,0

            MOVWF DIVIDENDO_H
            MOVF  DEFUZZ_DEN_L,0
            MOVWF DIVIDENDO_L
            MOVLW 0X64
            MOVWF DIVISOR

            CALL  SUBRUTINA_DIVISION
            MOVF  COCIENTE_L,0
            MOVWF DIVISOR_TEMP

            CALL  DIVISIONSOTA
            MOVF  COCIENTE_H,0
            MOVWF DIVIDENDO_H
            MOVF  COCIENTE_L,0
            MOVWF DIVIDENDO_L
            MOVF  DIVISOR_TEMP,0
            MOVWF DIVISOR
            CALL  SUBRUTINA_DIVISION
            RETURN

DEFUZIFICAR2

            MOVF  OUT2P1,0
            MOVWF LIMITE1
            MOVF  OUT2P2,0
            MOVWF LIMITE2
            MOVF  OUT2P3,0
            MOVWF LIMITE3
            MOVF  OUT2P4,0
            MOVWF LIMITE4
            MOVF  OUT2P5,0
            MOVWF LIMITE5

            CALL  DEFUZIFICAR

            CALL  CENTROIDE
            MOVF  COCIENTE_L,0
            MOVWF RES_FUERZA
```

```
            RETURN

;-------------------------------------------------------------------------------

;-------------------------------------------------------------------------------

;********************
SUBRUTINA_SUSTITUCION
;********************

; ESTA SUBRUTINA, EFECTUA LA SUSTITUCIÓN DE UN VALOR QUE ESTA DADO EN EL
; REGISTRO "SUBSTITUIR" EN EL REGISTRO A DONDE APUNTE EL PUNTERO DE
; DIRECCIONAMIENTO INDIRECTO (FSR), SI Y SÓLO SI EL PRIMERO ES MAYOR QUE
; EL SEGUNDO

            MOVF  SUBSTITUIR,0
            SUBWF INDF,0
            BTFSC STATUS,C

            RETURN

            MOVF  SUBSTITUIR,0
            MOVWF INDF

            RETURN

;-------------------------------------------------------------------------------

;-------------------------------------------------------------------------------

;***************
SUBRUTINA_MINIMO
;***************

; ESTA SUBRUTINA, ENCUENTRA EL MÍNIMO DE DOS VALORES DADOS EN LOS REGISTROS
; "MIN_1" Y "MIN_2" Y LO DEVUELVE EN EL REGISTRO "SUBSTITUIR"

            MOVF  MIN_1,0
            SUBWF MIN_2,0

            BTFSC STATUS,C
            CALL  MIN_SUBRUT_1
            BTFSS STATUS,C
            CALL  MIN_SUBRUT_2

            RETURN

MIN_SUBRUT_1

            MOVF  MIN_1,0
            MOVWF SUBSTITUIR
```

```
            RETURN

MIN_SUBRUT_2

            MOVF  MIN_2,0
            MOVWF SUBSTITUIR

            RETURN

;-------------------------------------------------------------------------------

;-------------------------------------------------------------------------------

;*****************
SUBRUTINA_CALC_MEM
;*****************

; ESTA SUBRUTINA, REALIZA EL CÁLCULO DE LA MEMBRESIA DANDO LOS PUNTOS BASE
; DEL CLUSTER TRIANGULAR (TRIANGULO RECTANGULO; ES DECIR, DIVIDIDO EN SU ALTURA)
; SIENDO "P1" EL PUNTO IZQUIERDO Y "P2" EL PUNTO DERECHO Y SI ES LA PARTE CON
; PENDIENTE POSITIVA (BIT "PENDIENTE_POS" DEL REGISTRO "FUZZY" = 1) O CON
; PENDIENTE NEGATIVA (BIT "PENDIENTE_POS" DEL REGISTRO "FUZZY" = 0) DEL VALOR
; DADO EN EL REGISTRO "X" (COMPRENDIDO ENTRE "P1" Y  "P2") DANDO EL
; RESULTADO EN EL REGISTOR "MEMBRESIA"

; SE RELAIZA EL CÁLCULO DEL DENOMINADOR Y SE DECIDE EL CASO DE PENDIENTE
; POSITIVA O NEGATIVA

            MOVF  P1,0
            SUBWF P2,0
            MOVWF DEN

            BTFSC FUZZY_REG,PENDIENTE_POS
            CALL  MEM_SUBRUT_1
            BTFSS FUZZY_REG,PENDIENTE_POS
            CALL  MEM_SUBRUT_2

; SE CALCULA LA MEMBRESIA, USANDO LAS SUBRUTINAS DE MULTIPLICACION Y DIVISION

            MOVLW 0X64
            MOVWF MULT_2
            MOVF  NUM,0
            MOVWF MULT_1_L
            CALL  SUBRUTINA_MULTIPLICACION

            MOVF  RES_MULT_L,0
            MOVWF DIVIDENDO_L
            MOVF  RES_MULT_H,0
            MOVWF DIVIDENDO_H
            MOVF  DEN,0
            MOVWF DIVISOR
```

```
            CALL  SUBRUTINA_DIVISION

            MOVF  COCIENTE_L,0
            MOVWF MEMBRESIA

            RETURN

MEM_SUBRUT_1

; SUBRUTINA QUE CALCULA EL NUMERADOR PARA EL CASO DE PENDIENTE NEGATIVA

            MOVF  X,0
            SUBWF P2,0
            MOVWF NUM

            RETURN

MEM_SUBRUT_2

; SUBRUTINA QUE CALCULA EL NUMERADOR PARA EL CASO DE PENDIENTE POSITIVA

            MOVF  P1,0
            SUBWF X,0
            MOVWF NUM

            RETURN

;-------------------------------------------------------------------------------

;-------------------------------------------------------------------------------

;***********************
SUBRUTINA_MULTIPLICACION
;***********************

; ESTA SUBRUTINA REALIZA LA MULTIPLICACION DE DOS NUMEROS
; DE 8 BITS, DADOS EN LOS REGISTROS "MULT_1_L" Y "MULT_2"
; DANDO UN RESULTADO DE 16 BITS EN LOS REGISTROS "RES_MULT_L"
; Y "RES_MULT_H" (L = PARTE MENOS SIGNIFICATIVA, H = PARTE MAS
; SIGNIFICATIVA)

; INICIALIZACION DE PARAMETROS PARA LA MULTIPLICACION

            CLRF  MULT_1_H
            CLRF  RES_MULT_H
            CLRF  RES_MULT_L
            CLRF  CONTADOR
            CLRF  COMPARADOR_MULT
            INCF  COMPARADOR_MULT,1


CICLO_MULT
```

```
; CICLO PRINCIPAL DEL CUERPO DE LA MULTIPLICACION

            MOVF  COMPARADOR_MULT,0
            ANDWF MULT_2,0
            BTFSS STATUS,Z
            CALL  MULT_SUBRUT_1

            BCF         STATUS,C
            RLF         MULT_1_L,1
            RLF         MULT_1_H,1
            RLF         COMPARADOR_MULT,1
            INCF  CONTADOR,1
            BTFSS CONTADOR,3
            GOTO  CICLO_MULT

            RETURN

MULT_SUBRUT_1

; SUBRUTINA QUE REALIZA LA SUMA DENTRO DE LA MULTIPLICACION

            MOVF  MULT_1_L,0
            ADDWF RES_MULT_L,1
            BTFSC STATUS,C
            INCF  RES_MULT_H,1
            MOVF  MULT_1_H,0
            ADDWF RES_MULT_H,1

            RETURN

;-------------------------------------------------------------------------------

;-------------------------------------------------------------------------------

;*****************
SUBRUTINA_DIVISION
;*****************

; ESTA SUBRUTINA LLEVA A CABO UNA DIVISIÓN DADO UN DIVIDENDO DE 16 BITS EN
; LOS REGISTROS "DIVIDENDO_H" (PARTE MÁS SIGNIFICATIVA) Y "DIVIDENDO_L" (PARTE
; MENOS SIGNIFICATIVA) ENTRE UN DIVISOR DE 8 BITS DADO EN EL REGISTRO "DIVISOR"
; DÁNDONOS UN RESULTADO DE 16 BITS EN LOS REGISTROS "COCIENTE_H" Y "COCIENTE_L"

; INICIALIZACION DE PARAMETROS PARA LA DIVISION

            CLRF  COCIENTE_L
            CLRF  COCIENTE_H
            CLRF  DIVIDENDO_AUX

            MOVLW 0X10
            MOVWF CONTADOR
```

```
CICLO_DIV

; CICLO PRINCIPAL DEL CUERPO DE LA DIVISION

            BCF         STATUS,C
            RLF         COCIENTE_L,1
            RLF         COCIENTE_H,1
            RLF         DIVIDENDO_L,1
            RLF         DIVIDENDO_H,1
            RLF         DIVIDENDO_AUX,1

            MOVF  DIVISOR,0
            SUBWF DIVIDENDO_AUX,0
            BTFSC STATUS,C
            CALL  DIV_SUBRUT_1

            DECF  CONTADOR,1
            BTFSS STATUS,Z
            GOTO  CICLO_DIV

            RETURN

DIV_SUBRUT_1

;SUBRUTINA QUE REALIZA LA RESTA AL DIVIDIR Y SUMA AL COCIENTE

            MOVWF DIVIDENDO_AUX
            INCF  COCIENTE_L,1

            RETURN

;--------------------------------------------------------------------------------

DEFUZIFICAR

            CLRF  MEM_1
            CLRF  MEM_2
            CLRF  MEM
            CLRF  DEFUZZ_DEN_L
            CLRF  DEFUZZ_DEN_H
            CLRF  DEFUZZ_NUM_L
            CLRF  DEFUZZ_NUM_H
            CLRF  DEFUZZ_NUM_H_2

DEFUZZ_1ER_CLUSTER

            MOVF  LIMITE1,0
            MOVWF P1
            MOVWF X
            MOVF  LIMITE2,0
            MOVWF P2
```

```
CICLO_1ER_CLUSTER

            BSF         FUZZY_REG,PENDIENTE_POS
            CALL  SUBRUTINA_CALC_MEM
            MOVF  MEMBRESIA,0
            MOVWF MIN_1
            MOVF  MEM_OUT_C1,0
            MOVWF MIN_2
            CALL  SUBRUTINA_MINIMO
            MOVF  SUBSTITUIR,0
            MOVWF MEM_1

            BCF         FUZZY_REG,PENDIENTE_POS
            CALL  SUBRUTINA_CALC_MEM
            MOVF  MEMBRESIA,0
            MOVWF MIN_1
            MOVF  MEM_OUT_C2,0
            MOVWF MIN_2
            CALL  SUBRUTINA_MINIMO

            MOVLW MEM_1
            MOVWF FSR
            CALL  SUBRUTINA_SUSTITUCION

            MOVF  MEM_1,0
            MOVWF MEM

            ADDWF DEFUZZ_DEN_L,1
            BTFSC STATUS,C
            INCF  DEFUZZ_DEN_H,1

            MOVF  MEM,0
            MOVWF MULT_1_L
            MOVF  X,0
            MOVWF MULT_2
            CALL  SUBRUTINA_MULTIPLICACION

            MOVF  RES_MULT_L,0
            ADDWF DEFUZZ_NUM_L,1
            BTFSC STATUS,C
            INCF  DEFUZZ_NUM_H,1
            MOVF  RES_MULT_H,0
            ADDWF DEFUZZ_NUM_H,1
            BTFSC STATUS,C
            INCF  DEFUZZ_NUM_H_2,1

            INCF  X,1
            MOVF  P2,0
            XORWF X,0
            BTFSC STATUS,Z
            GOTO  DEFUZZ_2DO_CLUSTER
```

```
            GOTO  CICLO_1ER_CLUSTER

DEFUZZ_2DO_CLUSTER

            MOVF  LIMITE2,0
            MOVWF P1
            MOVWF X
            MOVF  LIMITE3,0
            MOVWF P2

CICLO_2DO_CLUSTER

            BSF         FUZZY_REG,PENDIENTE_POS
            CALL  SUBRUTINA_CALC_MEM
            MOVF  MEMBRESIA,0
            MOVWF MIN_1
            MOVF  MEM_OUT_C2,0
            MOVWF MIN_2
            CALL  SUBRUTINA_MINIMO
            MOVF  SUBSTITUIR,0
            MOVWF MEM_1

            BCF         FUZZY_REG,PENDIENTE_POS
            CALL  SUBRUTINA_CALC_MEM
            MOVF  MEMBRESIA,0
            MOVWF MIN_1
            MOVF  MEM_OUT_C3,0
            MOVWF MIN_2
            CALL  SUBRUTINA_MINIMO

            MOVLW MEM_1
            MOVWF FSR
            CALL  SUBRUTINA_SUSTITUCION

            MOVF  MEM_1,0
            MOVWF MEM

            ADDWF DEFUZZ_DEN_L,1
            BTFSC STATUS,C
            INCF  DEFUZZ_DEN_H,1

            MOVF  MEM,0
            MOVWF MULT_1_L
            MOVF  X,0
            MOVWF MULT_2
            CALL  SUBRUTINA_MULTIPLICACION

            MOVF  RES_MULT_L,0
            ADDWF DEFUZZ_NUM_L,1
            BTFSC STATUS,C
            INCF  DEFUZZ_NUM_H,1
            MOVF  RES_MULT_H,0
```

```
            ADDWF DEFUZZ_NUM_H,1
            BTFSC STATUS,C
            INCF  DEFUZZ_NUM_H_2,1

            INCF  X,1
            MOVF  P2,0
            XORWF X,0
            BTFSC STATUS,Z
            GOTO  DEFUZZ_3ER_CLUSTER
            GOTO  CICLO_2DO_CLUSTER

DEFUZZ_3ER_CLUSTER

            MOVF  LIMITE3,0
            MOVWF P1
            MOVWF X
            MOVF  LIMITE4,0
            MOVWF P2

CICLO_3ER_CLUSTER

            BSF         FUZZY_REG,PENDIENTE_POS
            CALL  SUBRUTINA_CALC_MEM
            MOVF  MEMBRESIA,0
            MOVWF MIN_1
            MOVF  MEM_OUT_C3,0
            MOVWF MIN_2
            CALL  SUBRUTINA_MINIMO
            MOVF  SUBSTITUIR,0
            MOVWF MEM_1

            BCF         FUZZY_REG,PENDIENTE_POS
            CALL  SUBRUTINA_CALC_MEM
            MOVF  MEMBRESIA,0
            MOVWF MIN_1
            MOVF  MEM_OUT_C4,0
            MOVWF MIN_2
            CALL  SUBRUTINA_MINIMO

            MOVLW MEM_1
            MOVWF FSR
            CALL  SUBRUTINA_SUSTITUCION

            MOVF  MEM_1,0
            MOVWF MEM

            ADDWF DEFUZZ_DEN_L,1
            BTFSC STATUS,C
            INCF  DEFUZZ_DEN_H,1

            MOVF  MEM,0
            MOVWF MULT_1_L
```

```
            MOVF  X,0
            MOVWF MULT_2
            CALL  SUBRUTINA_MULTIPLICACION

            MOVF  RES_MULT_L,0
            ADDWF DEFUZZ_NUM_L,1
            BTFSC STATUS,C
            INCF  DEFUZZ_NUM_H,1
            MOVF  RES_MULT_H,0
            ADDWF DEFUZZ_NUM_H,1
            BTFSC STATUS,C
            INCF  DEFUZZ_NUM_H_2,1

            INCF  X,1
            MOVF  P2,0
            XORWF X,0
            BTFSC STATUS,Z
            GOTO  DEFUZZ_4TO_CLUSTER
            GOTO  CICLO_3ER_CLUSTER

DEFUZZ_4TO_CLUSTER

            MOVF  LIMITE4,0
            MOVWF P1
            MOVWF X
            MOVF  LIMITE5,0
            MOVWF P2

CICLO_4TO_CLUSTER

            BSF         FUZZY_REG,PENDIENTE_POS
            CALL  SUBRUTINA_CALC_MEM
            MOVF  MEMBRESIA,0
            MOVWF MIN_1
            MOVF  MEM_OUT_C4,0
            MOVWF MIN_2
            CALL  SUBRUTINA_MINIMO
            MOVF  SUBSTITUIR,0
            MOVWF MEM_1

            BCF         FUZZY_REG,PENDIENTE_POS
            CALL  SUBRUTINA_CALC_MEM
            MOVF  MEMBRESIA,0
            MOVWF MIN_1
            MOVF  MEM_OUT_C5,0
            MOVWF MIN_2
            CALL  SUBRUTINA_MINIMO

            MOVLW MEM_1
            MOVWF FSR
            CALL  SUBRUTINA_SUSTITUCION
```

```
            MOVF  MEM_1,0
            MOVWF MEM

            ADDWF DEFUZZ_DEN_L,1
            BTFSC STATUS,C
            INCF  DEFUZZ_DEN_H,1

            MOVF  MEM,0
            MOVWF MULT_1_L
            MOVF  X,0
            MOVWF MULT_2
            CALL  SUBRUTINA_MULTIPLICACION

            MOVF  RES_MULT_L,0
            ADDWF DEFUZZ_NUM_L,1
            BTFSC STATUS,C
            INCF  DEFUZZ_NUM_H,1
            MOVF  RES_MULT_H,0
            ADDWF DEFUZZ_NUM_H,1
            BTFSC STATUS,C
            INCF  DEFUZZ_NUM_H_2,1

            INCF  X,1
            MOVF  P2,0
            XORWF X,0
            BTFSC STATUS,Z
            RETURN
            GOTO  CICLO_4TO_CLUSTER

;-------------------------------------------------------------------------------
;-------------------------------------------------------------------------------

DIVISIONSOTA

            CLRF  COCIENTE_L
            CLRF  COCIENTE_H
            CLRF  DIVIDENDO_AUX

            MOVLW 0X18
            MOVWF CONTADOR

CICLO_DIVISIONSOTA

            BCF         STATUS,C
            RLF         COCIENTE_L,1
            RLF         COCIENTE_H,1
            RLF         DEFUZZ_NUM_L,1
            RLF         DEFUZZ_NUM_H,1
            RLF         DEFUZZ_NUM_H_2,1
            RLF         DIVIDENDO_AUX,1

            MOVF  DIVISOR,0
```

```
            SUBWF DIVIDENDO_AUX,0
            BTFSC STATUS,C
            CALL  DIVISIONSOTA_SUBRUT_1

            DECF  CONTADOR,1
            BTFSS STATUS,Z
            GOTO  CICLO_DIVISIONSOTA

            RETURN

DIVISIONSOTA_SUBRUT_1

            MOVWF DIVIDENDO_AUX
            INCF  COCIENTE_L,1

            RETURN

;-------------------------------------------------------------------------------
;-------------------------------------------------------------------------------
;-------------------------------------------------------------------------------
;-------------------------------------------------------------------------------

;*******************************************************************
;*                     RUTINA DE ATENCIÓN A INTERRUPCIÓN           *
;*******************************************************************

            ORG   0X3F0

ISR

            MOVWF W_TEMP                                    ; SE SALVAN LAS VARIABLES DE CONTEXTO
            MOVF  STATUS,0
            MOVWF STATUS_TEMP
            BCF         STATUS,RP0

            BTFSS INTCON,T0IE                           ; DETERMINA SI SE TRATA DE UNA INTERRUPCIÓN
            GOTO  NO_INT_TMR0                           ; DEL TIMER0 O DE OTRO PERIFÉRICO

            BTFSC INTCON,T0IF
            GOTO  INT_TMR0

NO_INT_TMR0

            BTFSC PIR1,TMR1IF                           ; CUANDO NO SE TRATA DE INTERRUPCIÓN DEL TIMER0
            GOTO  REESTABLECER_SISTEMA              ; DETERMINA SI FUE EL TIMER1

            BTFSC INTCON,RBIF                           ; LA INTERRUPCIÓN EN CAMBIO
            GOTO  PASO1_LECTURA_POTS

            BTFSC PIR1,ADIF                             ; O LA INTERRUPCIÓN DEL CONVERTIDOR A/D
            GOTO  INT_CONVERTIDOR
```

```
INT_TMR0

            BTFSC ALFA,TMR0_PASOS                       ; SI SE TRATA DEL TIMER0, DETERMINA SE INDICA
            GOTO  TEMPO                                 ; EL TIEMPO DE RETARDO DEL MOVIMIENTO DE LOS MOTORES

            BTFSC ALFA,TMR0_AD                          ; O EL TIEMPO DE CARGA DEL CAPACITOR DEL CONVERTIDOR
A/D
            GOTO  PASO2_LECTURA_POTS

INT_CONVERTIDOR

            BTFSC ALFA,SEGUNDA_CONVERSION           ; SIENDO LA INTERRUPCIÓN DEL ADC, DETERMINA SI ES
            GOTO  CONVERSION_DOS                        ; LA PRIMERA O SEGUNDA CONVERSIÓN
            GOTO  CONVERSION_UNO

CONVERSION_UNO

            MOVF  ADRESH,0                              ; DURANTE LA PRIMERA CONVERSIÓN, LEE EL RESULTADO DE
ÉSTA
            MOVWF VALOR_PRIMER_POT                  ; LO ALMACENA EN EL REGISTRO "VALOR_PRIMER_POT" E INICIA
            BCF         PIR1,ADIF                           ; LA SEGUNDA CONVERSIÓN

            BSF         STATUS,RP0
            BCF         PIE1,ADIE
            BCF         STATUS,RP0

            MOVLW 0X89
            MOVWF ADCON0
            BSF         ALFA,SEGUNDA_CONVERSION
            GOTO  PASO3_LECTURA_POTS

CONVERSION_DOS

            MOVF  ADRESH,0                              ; DURANTE LA SEGUNDA CONVERSIÓN, LEE EL RESULTADO DE
ÉSTA
            MOVWF VALOR_SEGUNDO_POT                 ; LO ALMACENA EN EL REGISTRO "VALOR_SEGUNDO_POT" Y
DESACTIVA
            MOVLW 0X80                                  ; LA INTERRUPCIÓN Y PROCEDIMIENTOS RELACIONADOS CON EL
ADC
            MOVWF ADCON0                                ; ACTIVA EL INDICADOR DE QUE CONTINÚA EL
CONTROLADOR DIFUSO
            BCF         PIR1,ADIF
            BSF         STATUS,RP0
            BCF         PIE1,ADIE
            BCF         STATUS,RP0
            BCF         ALFA,SEGUNDA_CONVERSION
            BCF         ALFA,TMR0_AD
            BCF         ADCON0,ADON
            BSF         ALFA,FUZZY
            GOTO  RETURN_ISR

PASO1_LECTURA_POTS
```

```
            BSF         ADCON0,ADON                     ; ACTIVA EL ADC, DESACTIVA Y LIMPIA LA
INTERRUPCIÓN EN CAMBIO
            BCF         INTCON,RBIF                     ; PROGRAMA EL TIEMPO DE ESPERA NECESARIO PARA
CARGAR EL
            BCF         INTCON,RBIE                     ; CAPACITOR DE MUESTREO Y RETENCIÓN

            MOVLW 0XC0
            MOVWF TMR0
            BCF         INTCON,T0IF
            BSF         INTCON,T0IE
            BSF         ALFA,TMR0_AD

            GOTO  RETURN_ISR

PASO2_LECTURA_POTS

            BCF         INTCON,T0IE                     ; TRANSCURRIDO EL TIEMPO NECESARIO DE MUESTREO Y
RETENCIÓN,
            BCF         PIR1,ADIF                       ; ACTIVA EL CONVERTIDOR A/D

            BSF         STATUS,RP0
            BSF         PIE1,ADIE

            BCF         STATUS,RP0
            BSF         ADCON0,GO_DONE

            GOTO  RETURN_ISR

PASO3_LECTURA_POTS

            MOVLW 0XC0                          ; PROGRAMA EL TIEMPO DE ESPERA NECESARIO PARA CARGAR
EL
            MOVWF TMR0                          ; CAPACITOR DE MUESTREO Y RETENCIÓN, EN ESTA OCASIÓN
PARA
            BCF         INTCON,T0IF                     ; LA SEGUNDA CONVERSIÓN
            BSF         INTCON,T0IE
            GOTO  RETURN_ISR

IZQUIERDA

            MOVF  ULTIMO_PASO,0                 ; SI EL MOTOR SE ESTÁ MOVIENDO HACIA LA IZQUIERDA,
            XORLW 0X0A                          ; DETERMINA CUÁL ES EL SIGUIENTE PASO A MANDAR Y SALTA
A
            BTFSC STATUS,Z                      ; EJECUTARLO
            GOTO  PASO2

            MOVF  ULTIMO_PASO,0
            XORLW 0X09
            BTFSC STATUS,Z
            GOTO  PASO3
```

```
            MOVF  ULTIMO_PASO,0
            XORLW 0X05
            BTFSC STATUS,Z
            GOTO  PASO4
            GOTO  PASO1

DERECHA

            MOVF  ULTIMO_PASO,0                     ; SI EL MOTOR SE ESTÁ MOVIENDO HACIA LA DERECHA,
            XORLW 0X0A                              ; DETERMINA CUÁL ES EL SIGUIENTE PASO A MANDAR Y SALTA
A
            BTFSC STATUS,Z                          ; EJECUTARLO
            GOTO  PASO4

            MOVF  ULTIMO_PASO,0
            XORLW 0X09
            BTFSC STATUS,Z
            GOTO  PASO1

            MOVF  ULTIMO_PASO,0
            XORLW 0X05
            BTFSC STATUS,Z
            GOTO  PASO2
            GOTO  PASO3

PASO1

            BTFSC ALFA,PRIMER_PASO                  ; MANDA EL PASO 1 AL PUERTO CORRESPONDIENTE (SEGÚN EL
MOTOR)
            CALL  PRIMERA_VEZ
            MOVLW 0X0A
            MOVWF INDF
            MOVWF ULTIMO_PASO
            GOTO  RETURN_ISR

PASO2

            BTFSC ALFA,PRIMER_PASO                  ; MANDA EL PASO 2 AL PUERTO CORRESPONDIENTE (SEGÚN EL
MOTOR)
            CALL  PRIMERA_VEZ
            MOVLW 0X09
            MOVWF INDF
            MOVWF ULTIMO_PASO
            GOTO  RETURN_ISR

PASO3

            BTFSC ALFA,PRIMER_PASO                  ; MANDA EL PASO 3 AL PUERTO CORRESPONDIENTE (SEGÚN EL
MOTOR)
            CALL  PRIMERA_VEZ
            MOVLW 0X05
            MOVWF INDF
```

```
            MOVWF ULTIMO_PASO
            GOTO  RETURN_ISR

PASO4

            BTFSC ALFA,PRIMER_PASO                  ; MANDA EL PASO 4 AL PUERTO CORRESPONDIENTE (SEGÚN EL
MOTOR)
            CALL  PRIMERA_VEZ
            MOVLW 0X06
            MOVWF INDF
            MOVWF ULTIMO_PASO
            GOTO  RETURN_ISR

PRIMERA_VEZ

            BCF         INTCON,T0IF                     ; SI ES LA PRIMERA VEZ QUE ENTRA A LA SECUENCIA
DE ENVIAR PASOS,
            BSF         INTCON,T0IE                     ; PROGRAMA EL TIMER0 PARA QUE DE LA PAUTA ENTRE
ELLOS, MEDIANTE
                                                              ; SU INTERRUPCIÓN Y ACTIVA UN INDICADOR
PARA INDICAR QUE EL TIMER0
            BSF         STATUS,RP0                      ; SE ENCUENTRA EN ESTA SECUENCIA
            MOVLW 0X05
            MOVWF OPTION_REG
            BCF         STATUS,RP0

            MOVLW 0X00
            MOVWF TMR0
            BSF         ALFA,TMR0_PASOS
            RETURN

TEMPO

            BCF         INTCON,T0IF                     ; SECUENCIA EJECUTADA ENTRE PASO Y PASO, QUE
LIMPIA LA INTERRUPCIÓN
            BCF         ALFA,PRIMER_PASO            ; DEL TIMER0 Y LA PREPARA PARA EL SIGUIENTE INTERVALO,
JUZGA SI YA
            DECF  CONT_PASOS_L,1                    ; SE ENVIARON LOS PASOS DEBIDOS, DE SER ASÍ VA A LAS
RUTINAS
            BTFSC STATUS,Z                          ; CORRESPONDIENTES Y SI NO, SIGUE MANDANDO PASOS
            CALL  TERMINA_CONT_PASOS_L
            BTFSC GAMMA,0
            GOTO  TERMINAN_PASOS
            BTFSS ALFA,SENTIDO
            GOTO  DERECHA
            GOTO  IZQUIERDA

TERMINA_CONT_PASOS_L

            DECF  CONT_PASOS_H,1                    ; SI TERMINAN LOS PASOS DEL CONTADOR BAJO, DECREMENTA
EL CONTADOR
```

```
            BTFSC STATUS,Z                          ; ALTO, SI ÉSTE NO TERMINA, REINICA EL CONTADOR BAJO Y
VUELVE A
            BSF         GAMMA,0                             ; SEGUIR MANDANDO LOS PASOS RESTANTES.
            MOVLW 0X0B                              ; SI SE TERMINARON YA TODOS LOS PASOS, SI ES DEL MOTOR
1, MANDA
            MOVWF CONT_PASOS_L                      ; EJECUTAR LA MISMA SECUENCIA PERO PARA EL MOTOR 2. SI
ES ESTE
            RETURN                                          ; ÚLTIMO EL QUE TERMINO, ENTONCES SUSPENDE
LA ASIGNACIÓN DEL

INDICADORES NECESARIOS PARA                             ; TIMER0 A ESTA TAREA Y ACTIVA LOS
TERMINAN_PASOS                                      ; EJECUTAR EL TIRO, O EN EL CASO DE HABER ESTADO
REESTABLECIENDO EL

                                                        ; SISTEMA, ACTIVAR LOS INDICADORES DE QUE
ESTA TAREA YA TERMINÓ
            BTFSC ALFA,MOTOR2
            GOTO  TERMINAN_PASOS_M2
            BCF         GAMMA,0
            BCF         ALFA,TMR0_PASOS
            BSF         ALFA,MOTOR2
            MOVF  ULTIMO_PASO,0
            MOVWF ULTIMO_PASO_M1
            MOVF  ULTIMO_PASO_M2,0
            MOVWF ULTIMO_PASO
            MOVLW PORTD
            MOVWF FSR
            BSF         ALFA,MOTORES

;PROBABLEMENTE OMITIDO CON EL CONTROLADOR DIFUSO
;*****************************************
            MOVLW 0X01
            MOVWF CONT_PASOS_H
            MOVF  PASOS_MOTOR_2,0
            MOVWF CONT_PASOS_L
;*****************************************
            GOTO  RETURN_ISR

TERMINAN_PASOS_M2

            BCF         GAMMA,0
            BCF         ALFA,TMR0_PASOS
            BCF         ALFA,MOTOR2
            MOVF  ULTIMO_PASO,0
            MOVWF ULTIMO_PASO_M2
            MOVF  ULTIMO_PASO_M1,0
            MOVWF ULTIMO_PASO
            MOVLW PORTB
            MOVWF FSR
            BTFSS BETA,REESTABLECER
            BSF         BETA,TIRO
            BTFSC BETA,REESTABLECER
            CALL  REESTABLECIDO
```

```
            MOVLW 0X01
            MOVWF CONT_PASOS_H
            MOVLW 0X0B
            MOVWF CONT_PASOS_L
            GOTO  RETURN_ISR

REESTABLECER_SISTEMA


            MOVF  PORTB,0                                   ; PARA REESTABLECER EL SISTEMA, INVIERTE LA
DIRECCIÓN DE GIRO
            BCF         T1CON,TMR1ON                        ; DE LOS MOTORES Y MANDA A EJECUTAR LA SECUENCIA
DE SU MOVIMIENTO
            BCF         PIR1,TMR1IF                         ; NUEVAMENTE
            BSF         STATUS,RP0
            BCF         PIE1,TMR1IE
            BCF         STATUS,RP0
            BCF         PORTA,5

            MOVF  PASOS_MOTOR_1,0
            MOVWF CONT_PASOS_H

            MOVLW 0X03
            XORWF BETA,1

            BSF         ALFA,MOTORES
            BSF         BETA,REESTABLECER

            GOTO  RETURN_ISR

REESTABLECIDO

            BCF         INTCON,T0IE                         ; INDICA QUE EL SISTEMA HA SIDO REESTABLECIDO,
PARA QUE EL SISTEMA
            BCF         BETA,REESTABLECER                   ; OPERATIVO LO CONOZCA Y REINICIE INTERNAMENTE A SUS
VALORES INICIALES
            BSF         BETA,SYSTEM_OK
            RETURN

RETURN_ISR

            MOVF  STATUS_TEMP,0                             ; REESTABLECER VARIABLES DE CONTEXTO
            MOVWF STATUS
            MOVF  W_TEMP,0
            RETFIE

;-------------------------------------------------------------------------------
;-------------------------------------------------------------------------------
;-------------------------------------------------------------------------------
;-------------------------------------------------------------------------------
```

END