

Apéndice B

Programa en Ensamblador para el PIC 16F877 de Microchip Corp.

```
;*****UDLAP*****
;PROGRAMA : com232.asm
;Función : Establece la rutina de los motores + el PWM integrado
;          con desplazamiento de 100 mm + algoritmo sensor
;          Serial port config: 9600 baudios,8 bits, paridad = none
;
; Revisión : 1.00   14/05/05   Programa para : PIC16F877
; CPU Clock : 4 MHz           Reloj instrucción :
; WDT : Deshabilitado        Tipo de reloj : XT
; Code Prot : OFF
;*****GiovanniSandre*****
```

```
LIST P=16F877 ;Se indica el modo de procesador
RADIX HEX ;Sistema de numeración hexadecimal
#include <P16f877.inc> ;se incluye la definición de los
;variables
```

```
CBLOCK 0x20
OUTER ;retrazo de W ms
INNER ;rutina contador 1ms
```

```
BITCOUNT; Contador de 8 bits
CURRENT ; salida de datos capturados 8 bits rlf
CONT0 ;contador de 0 a 255 pulsos para el fin de recorrido Y
CONT1 ;contador de 0 a 255 pulsos para el fin de recorrido X
CONT2 ;contador que sirve para comparar la dirección del recorrido en Y
DataByte1
ENDC
```

```
;<<<<<<----- VECTOR DE RESET ----->>>>>>
ORG 0x00 ;Inicio en el Vector de Reset
nop ;implementado por que el programa se salta la
; primera instrucción
goto INICIO ; Va a la primera instrucción del programa
```

```
;<<<<<<----- VECTOR DE INTERRUPCION ----->>>>>>
ORG 0x04 ;Vector de interrupción
btfsc PORTD,6 ;rutina de comparación para parar motor
goto STOPY ; o X o Y, dependiendo si esta aprendido el motor Y
goto STOPX
```

```
;<<<<<<----- INICIO PROGRAMA PRINCIPAL ----->>>>>>
INICIO
call PORTS
```

```
call    BAUDRATE
```

PORTS; Inicialización de puertos

```
bcf     STATUS,RP0
bcf     STATUS,RP1
clrf   PORTC
clrf   PORTD
clrf   TMR1L
clrf   TMR1H
clrf   CONT0      ;contador de 255 para desplazamiento X
clrf   CONT1      ;contador de 255 para desplazamiento Y
clrf   CONT2      ;contador de dirección 0-derecha 1-izquierda
BANKSEL TRISC
```

```
movlw   B'10000011' ; importante control RC0-pulsos codificador, RC7-RX,
; RC6-TX, RC5-DirY
```

```
movwf   TRISC      ; RC4-control Y, RC3-control X, inicializo TMR1,
; RC2-PWM, RC1-none
```

```
BANKSEL PORTD
```

```
movlw   B'00000001' ; inicializar
```

```
movwf   PORTD      ; bit detector high (dectector off)
```

```
BANKSEL TRISD
```

```
movlw   B'00000010' ; RD7-X, RD6-Y
```

```
movwf   TRISD
```

```
BANKSEL PORTB
```

```
CLRF    PORTB      ; Initialize PORTB by clearing output
; data latches
```

```
BANKSEL TRISB      ; Select Bank 1
```

```
MOVLW   B'00000000' ; Value used to initialize data direction
```

```
MOVWF   TRISB
```

```
return
```

PWM; Inicialización del PWM

```
clrf   STATUS
```

```
clrf   CCP1CON      ; CCP Module esta apagado
```

```
clrf   TMR2         ; Limpiar Timer2
```

```
movlw   B'11001111' ; = 207, Decimal prescalador 16, 4MHz cristal, Frec de
; 300Hz
```

```
BANKSEL PR2
```

```
movwf   PR2         ;  $(0.003/(4*2.5*10^{-7}*16))-1 = 207.3125$ 
```

```
movlw   B'00100101' ;  $0010010110 = 0.0006/(2.5*10^{-7}*16) = 150$ 
```

```
BANKSEL CCPR1L
```

```
movwf   CCPR1L      ; Ciclo de Trabajo es del 20% del Periodo PWM
```

```
clrf   INTCON       ; Habilitar interrupciones y limpiar T0IF
```

```
BANKSEL PIE1
```

```
clrf   PIE1         ; Habilitar interrupciones periféricas
```

```
BANKSEL PIR1
```

```

clrf      PIR1          ; Limpiar interrupciones banderas periféricas
movlw    B'00101100'   ; PWM , 2 LSbs del ciclo de trabajo = 10
movwf    CCP1CON
BSF      T2CON, TMR2ON ; Timer2 starts to increment
BSF      T2CON, T2CKPS1 ; Prescalade 16

```

MOTORS ; rutina de motores

```

BANKSEL  PIE1
bsf      PIE1, TMR1IE   ; habilitar TMR1 Overflow Interrupt
bsf      INTCON, PEIE   ; habilitar

```

```

BANKSEL  T1CON
movlw    B'00000011'    ; TMR1 aprendido, reloj externo, no
                        ; sincronizado
movwf    T1CON
clrf     PORTC

```

APRENDEY; prende motor Y

```

bsf      INTCON, GIE
btfss   CONT2, 0
goto    ida
goto    regreso ;rutinas de corrección de error " ida y regreso "
ida
movlw    B'10010011'    ;364 puntos 1/4 de vuelta 1mm hacia + x
movwf    TMR1L
movlw    B'11111110'
movwf    TMR1H
goto    skip
regreso
movlw    B'10001101'    ;370 puntos 1/4 de vuelta 1mm hacia - x
movwf    TMR1L
movlw    B'11111110'
movwf    TMR1H
skip
bsf      PORTC, 4        ;prende control de X decoder
bsf      PORTD, 6        ;prende motor X con RC6
incf    CONT0           ;contador de 255 pulsos fin de recorrido X

```

RUTINAY; rutina de espera a interrupción

```

goto    RUTINAX

```

STOPY; rutina de apagado de motor Y

```

bcf     PORTD, 6
bcf     PIR1, TMR1IF
call    SENSOR

```

REINICIO; ciclo de reinicio de motores

```

bcf          STATUS,2    ; resta 255-CONT0 si es 1 status prende X
                ; sino regresa a la rutina
                ; hasta que se acabe el desplazamiento

movlw       D'100'
subwf      CONT0,0
btfss     STATUS,2
goto      APRENDEY
bcf       PORTC,4    ; apaga control de X codificador
call     DIR

```

APRENDEX; rutina de encendido del motor X con RC7

```

bsf        INTCON,GIE
movlw     B'01011101' ; 418p 1/4 de vuelta 1mm
movwf    TMR1L
movlw     B'11111110'
movwf    TMR1H
bsf       PORTC,3    ; prende control de X codificador
bsf       PORTD,7
incf     CONT1      ; contador de paro para el total del desplazamiento
bcf       STATUS,2
movlw     B'01100100'
subwf    CONT1,0
btfsc   STATUS,2
goto     bucle

```

STOPX ;rutina de apagado de motor Y de 418 pulsos y 255 de largo

```

bcf       PORTD,7
bcf       PIR1,TMR1IF
clrf     CONT0      ;limpía CONT0 para que realice los 255 y no +
bcf       PORTC,3    ;apaga control de Y decoder
bcf       STATUS,2
movlw     B'01100100'
subwf    CONT1,0
btfsc   STATUS,2
goto     STOPY
goto     APRENDEX

```

bucle ;bucle de paro para el ultimo desplazamiento 255 de Y y regreso al punto 0,0

```

bcf       PORTD,7
nop      ;delay de conmutación
nop
nop
bsf       INTCON,GIE
movlw     B'10101111' ; 50000 -10cm
movwf    TMR1L
movlw     B'00111100'
movwf    TMR1H

```

```

    bsf      PORTD,4      ;cambia dirección
    bsf      PORTC,3      ;prende control de Y decoder
    bsf      PORTD,7
    goto     RUTINAX

DIR; rutina de 0 a 1 y de 1 a 0, el bit 5 para dirección
    incf     CONT2
    btfsc   CONT2,0
    bsf     PORTC,5
    btfss   CONT2,0
    bcf     PORTC,5
    return

SENSOR
    BANKSEL PORTD
    movlw   B'00000001'   ; inicializar
    movwf   PORTD         ; bit detector high (detector off)
    bsf     STATUS, RP0
    movlw   B'00000010'   ; RD0-Vin, RD1-Vout del sensor
    movwf   TRISD

main_loop
    movlw   D'5'         ; = OUTER
    call    delay         ; detector
    call    gp2d02       ; leer dato

delay_1_ms ; rutina de mas de 1ms de tiempo de reloj
    movlw   0xC7         ; se inicializa
    movwf   INNER       ; contador
inner_loop
    nop     ; 5 cinco ciclos (loop)
    nop
    decfsz INNER,f
    goto    inner_loop
    return

delay ;rutina de (W) ms
    bcf     PORTD,0
    movwf   OUTER       ; loop de (w) times
outer_loop
    call    delay_1_ms   ; cada 1ms
    decfsz OUTER,f
    goto    outer_loop
    return

; gp2d02 subrutina
; Esta rutina lee el dato del GP2D02 y almacena el resultado

```

; en el W register de regreso

gp2d02

nop
nop

wait_for_reading

btfss PORTD, 1 ; espera hasta que este hecha la medición
goto wait_for_reading

bsf PORTD, 0 ; apaga detector(punto)
clrf CURRENT ; limpiar valor anterior

movlw D'8' ; inicializar reloj
movwf BITCOUNT ; contador de 8 bits

bcf STATUS, C ; asegurar acarreo
; limpia el bit para la rotación
nop ; retraso de reloj

read_bit

bcf PORTD, 0 ; prende sensor
nop ; retraso de reloj
nop

rlf CURRENT, f ; rota bit.

btfsc PORTD, 1 ; checa el bit
bsf CURRENT, 0 ; uno si la salida es 1
bsf PORTD, 0 ; apagas ensor

nop ; retrasos
nop

decfsz BITCOUNT, f ; saber si son los ocho bits
goto read_bit

movf CURRENT,0 ; sacar el valor a W para posteriormente
; mandarlo al buffer de salida del comm

movwf PORTB
goto TRANSMIT

BAUDRATE; inicalización del comm

BANKSEL SPBRG ; selección del Baud Rate Generator Register
; en decimal para

movlw D'25' ; un cristal de 4MHz y un mínimo de error

movwf SPBRG

BANKSEL TXSTA

bsf TXSTA, BRGH ; selección del Baud Rate

```
bcf      TXSTA,SYNC ; forma asíncrona
BANKSEL RCSTA
bsf      RCSTA,SPEN
```

SETUPASYNTRANS; inicialización de la transmisión

```
BANKSEL PIE1
bcf      PIE1,TXIE
BANKSEL TXSTA
bsf      TXSTA,TXEN
```

SETUPSYNREC; inicialización de a recepción

```
BANKSEL RCSTA
bsf      RCSTA,CREN
```

WAITRX; rutina de espera de recepción

```
btfs    PIR1,RCIF
goto    WAITRX
```

RECEIVE; rutina de recepción y almacenamiento

```
BANKSEL RCREG
movf    RCREG,W ; registro de recepción
BANKSEL DataByte1 ; registro que contiene el caracter i
movwf   DataByte1 ; compara e inicializa el programa completo
movlw   B'01101001' ; sino se queda en la rutina y espera nuevo dato
bcf     STATUS,2
subwf   DataByte1,0
btfsc   STATUS,2
goto    PWM
goto    WAITRX
```

TRANSMIT; rutina de transmisión del dato o medición

```
BANKSEL CURRENT
movf    CURRENT,0
BANKSEL TXREG ; registro de transmisión
movwf   TXREG
goto    REINICIO
```

END