



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE INGENIERÍA

Laboratorio de Microcomputadoras

Práctica 5.
Control de actuadores

Grupo de Laboratorio: 4

Sanjuan Aldape Diana Paola
Grupo de Teoría: 5

Esparza Fuentes Jorge Luis
Grupo de Teoría: 4

Fecha de realización: 15 de marzo del 2022



DESARROLLO

Utilizando el circuito de potencia de motores de corriente directa y el sistema de desarrollo del microcontrolador PIC, realizar los programas solicitados.

1.- Considerando la asignación de terminales proporcionadas; realizar el programa que ejecute el control indicado en la siguiente tabla:

DATO Puerto Paralelo	ACCION	
	MOTOR M1	MOTOR M2
0x00	PARO	PARO
0x01	PARO	HORARIO
0x02	PARO	ANTI-HORARIO
0x03	HORARIO	PARO
0x04	ANTI-HORARIO	PARO
0x05	HORARIO	HORARIO
0x06	ANTI-HORARIO	ANTI-HORARIO
0x07	HORARIO	ANTI-HORARIO
0x08	ANTI-HORARIO	HORARIO

```
processor 16f877
include<pl6f877.inc>
;Constantes para comparar la combinación entrante
c0 equ H'00'
c1 equ H'01'
c2 equ H'02'
c3 equ H'03'
c4 equ H'04'
c5 equ H'05'
c6 equ H'06'
c7 equ H'07'
c8 equ H'08'
org 0
goto inicio
org 5
inicio:
clrf    PORTA      ;Limpia el puerto A
bsf     STATUS,RP0 ;Cambio al Banco 1
bcf     STATUS,RP1
movlw   H'00'      ;Mueve 0h a w
movwf   TRISB      ;Configura Puerto B como salida
movwf   TRISC      ;Configura Puerto C como salida
movlw   H'06'
movwf   ADCON1     ;Configura puertos A y E como digitales
movlw   H'3F'
movwf   TRISA      ;Configura el Puerto A como entrada
bcf     STATUS,RP0 ;Regresa al Banco 0
-----
```



```
clrf    PORTB        ;Limpia los bits de Puerto B (los apaga)
clrf    PORTC
movlw   H'06'
movwf   PORTC
ciclo:
;COMBINACION PARA MOTOR 1 Y 2 PARADOS
movlw   c0           ;Mueve 0 a w
xorwf   PORTA,w       ;Verifica si la entrada es 00
btfsc   STATUS,Z      ;Verifica el resultado de la xor
goto    CASO0         ;La combinacion es 00 y procede a la rutina
;COMBINACION PARA MOTOR 1 = PARO, MOTOR 2 = HORARIO
movlw   c1           ;Mueve 0 a w
xorwf   PORTA,w       ;Verifica si la entrada es 01
btfsc   STATUS,Z      ;Verifica el resultado de la xor
goto    CASO1         ;La combinacion es 01 y procede a la rutina
;COMBINACION PARA MOTOR 1 = PARO, MOTOR 2 = ANTIHORARIO
movlw   c2           ;Mueve 0 a w
xorwf   PORTA,w       ;Verifica si la entrada es 02
btfsc   STATUS,Z      ;Verifica el resultado de la xor
goto    CASO2         ;La combinacion es 02 y procede a la rutina
;COMBINACION PARA MOTOR 1 = HORARIO, MOTOR 2 = PARO
movlw   c3           ;Mueve 0 a w
xorwf   PORTA,w       ;Verifica si la entrada es 03
btfsc   STATUS,Z      ;Verifica el resultado de la xor
goto    CASO3         ;La combinacion es 03 y procede a la rutina
;COMBINACION PARA MOTOR 1 = ANTIHORARIO, MOTOR 2 = PARO
movlw   c4           ;Mueve 0 a w
xorwf   PORTA,w       ;Verifica si la entrada es 04
btfsc   STATUS,Z      ;Verifica el resultado de la xor
goto    CASO4         ;La combinacion es 04 y procede a la rutina
;COMBINACION PARA MOTOR 1 = HORARIO, MOTOR 2 = HORARIO
movlw   c5           ;Mueve 0 a w
xorwf   PORTA,w       ;Verifica si la entrada es 05
btfsc   STATUS,Z      ;Verifica el resultado de la xor
goto    CASO5         ;La combinacion es 04 y procede a la rutina
;COMBINACION PARA MOTOR 1 = ANTIHORARIO, MOTOR 2 = ANTIHORARIO
movlw   c6           ;Mueve 0 a w
xorwf   PORTA,w       ;Verifica si la entrada es 06
btfsc   STATUS,Z      ;Verifica el resultado de la xor
goto    CASO6         ;La combinacion es 04 y procede a la rutina
;COMBINACION PARA MOTOR 1 = HORARIO, MOTOR 2 = ANTIHORARIO
movlw   c7           ;Mueve 0 a w
xorwf   PORTA,w       ;Verifica si la entrada es 07
btfsc   STATUS,Z      ;Verifica el resultado de la xor
goto    CASO7         ;La combinacion es 04 y procede a la rutina
;COMBINACION PARA MOTOR 1 = ANTIHORARIO, MOTOR 2 = HORARIO
movlw   c8           ;Mueve 0 a w
xorwf   PORTA,w       ;Verifica si la entrada es 08
btfsc   STATUS,Z      ;Verifica el resultado de la xor
goto    CASO8         ;La combinacion es 05 y procede a la rutina
goto    ciclo         ;Vuelve a preguntar por una combinacion valida
```



```
CASO0:
    movlw    h'00'
    movwf    PORTB
    goto     ciclo
CASO1:
    movlw    h'08'
    movwf    PORTB
    goto     ciclo
CASO2:
    movlw    h'04'
    movwf    PORTB
    goto     ciclo
CASO3:
    movlw    h'01'
    movwf    PORTB
    goto     ciclo
CASO4:
    movlw    h'02'
    movwf    PORTB
    goto     ciclo
CASO4:
    movlw    h'02'
    movwf    PORTB
    goto     ciclo
CASO5:
    movlw    h'09'
    movwf    PORTB
    goto     ciclo
CASO6:
    movlw    h'06'
    movwf    PORTB
    goto     ciclo
CASO7:
    movlw    h'05'
    movwf    PORTB
    goto     ciclo
CASO8:
    movlw    h'0A'
    movwf    PORTB
    goto     ciclo
    end
```

2.- Realizar un programa que controle la cantidad de pasos que debe dar un motor así como el sentido de giro.

Dato Puerto Paralelo	Motor a pasos
0x00	Motor en paro
0x01	Gira en sentido horario durante 5 segundos
0x02	Gira en sentido anti horario por 10 segundos
0x03	Gira cinco vueltas en sentido horario
0x04	Gira 10 vueltas en sentido anti horario



```
PROCESSOR 16f877
INCLUDE <pl6f877.inc>
valor1 equ h'21'
valor2 equ h'22'
valor3 equ h'23'
CTE1 EQU H'01'
CTE2 EQU H'50'
CTE3 EQU H'60'
c0 equ h'00'
c1 equ h'01'
c2 equ h'02'
c3 equ h'03'
c4 equ h'04'
c5 equ h'05'
c6 equ h'06'
c7 equ h'07'
c8 equ h'08'
CONT equ 0x20
CONT2 equ 0x24
CTE4 EQU 0X03
CTE5 EQU 0XFF
REG4 EQU 0X25
REG5 EQU 0X26
NVUELTA EQU 0X27
    ORG 0
    GOTC INICIO
    ORG 5

INICIO:
    clrf PORTA
    BSF STATUS, RP0
    BCF STATUS, RP1
    MOVLW h'00'
    MOVWF TRISB
    MOVLW H'06'
    MOVWF ADCON1
    MOVLW H'3F'
    MOVWF TRISA
    BCF STATUS, RP0
    CLRF PORTB

LOOPMAIN:
    CLRF PORTB
    CALL RETARDO
    MOVF PORTA,W    ;w<--PORTA
    ADDWF PCL,F     ;PCL<--PORTA + PCL
    GOTC C0         ;PC+0
    GOTC C1         ;PC+1
    GOTC C2         ;PC+2
    GOTC C3         ;PC+3
    GOTC C4         ;PC+4

C0:
    MOVLW 0X00
    MOVWF PORTB
    GOTC LOOPMAIN

C1:
    CALL PASOS64
    GOTC LOOPMAIN

C2:
    CALL ANTI_PASOS64
    GOTC LOOPMAIN

C3:
    CLRF NVUELTA
    VUELTAS:
        CALL PASOS64
        INCF NVUELTA
        MOVE NVUELTA, W
        SUBLW 0X05
        BTFSS STATUS, Z
        GOTC VUELTAS
    GOTC LOOPMAIN

C4:
    CLRF NVUELTA
    VUELTA10:
        CALL ANTI_PASOS64
        INCF NVUELTA
        MOVE NVUELTA, W
        SUBLW 0X0A
        BTFSS STATUS, Z
        GOTC VUELTA10
    GOTC LOOPMAIN

PASOS64:
    CLRF CONT
    LOOP:
        CALL VUELTA
        INCF CONT
        MOVE CONT, W
        SUBLW 0X40
        BTFSS STATUS, Z
        GOTC LOOP
    FINLOOP:
        RETURN




VUELTA:
    CLRF CONT2
    LOOP1:
        CALL PASOS
        INCF CONT2
        MOVE CONT2, W
        SUBLW 0X08
        BTFSS STATUS, Z
        GOTC LOOP1
    FIN2:
        RETURN
```



```
PASOS:
    MOVLW H'C0'
    MOVWF PORTB
    CALL RETARDO
    MOVLW H'60'
    MOVWF PORTB
    CALL RETARDO
    MOVLW H'30'
    MOVWF PORTB
    CALL RETARDO
    MOVLW H'90'
    MOVWF PORTB
    CALL RETARDO
RETURN
ANTI_PASOS64:
    CLRF CONT
    LOOP2:
        CALL ANTI_VUELTA
        INCF CONT
        MOVF CONT, W
        SUBLW 0X40
        BTFSS STATUS, Z
        GOTC LOOP2
    FIN3:
        RETURN
ANTI_VUELTA:
    CLRF CONT2
    LOOP3:
        CALL ANTI_PASOS
        INCF CONT2
        MOVF CONT2, W
        SUBLW 0X08
        BTFSS STATUS, Z
        GOTC LOOP3
    FIN4:
        RETURN
-----

ANTI_PASOS:
    MOVLW H'90'
    MOVWF PORTB
    CALL RETARDO
    MOVLW H'30'
    MOVWF PORTB
    CALL RETARDO
    MOVLW H'60'
    MOVWF PORTB
    CALL RETARDO
    MOVLW H'C0'
    MOVWF PORTB
    CALL RETARDO
RETURN
RETARDO:
    MOVLW CTE1
    MOVWF valor1
tres:
    MOVLW CTE2
    MOVWF valor2
dos:
    MOVLW CTE3
    MOVWF valor3
uno:
    decfsz valor3
    goto uno
    decfsz valor2
    goto dos
    decfsz valor1
    goto tres
return
END ; fin del programa
```

3.- Utilizando un servo motor realizar el control mostrado en la siguiente tabla:

SW2	SW1	SW0	Posición Servo	Representación
1	0	0	Izquierda	 0°
0	1	0	Central	 90°
0	0	1	Derecha	 180°



```
PROCESSOR 16F877  
INCLUDE <pl6F877.inc>
```

```
valor1 equ h'21'  
valor2 equ h'22'  
valor3 equ h'23'  
cte1 equ h'01'  
cte2 equ h'50'  
cte3 equ h'60'  
c0 equ h'04'  
c1 equ h'02'  
c2 equ h'01'  
c3 equ h'03'  
cont equ 0x20  
cont2 equ 0x24  
regini equ 0x028  
regini2 equ 0x029  
reg1 equ 0x25  
reg2 equ 0x26  
nvuelta equ 0x27  
org 0  
goto INICIO  
org 5  
INICIO:  
    BCF STATUS, RP0  
    BCF STATUS, RP1  
    clrf PORTA  
    clrf PORTC  
    BSF STATUS, RP0  
    BCF STATUS, RP1  
    movlw 0x07  
    movwf ADCON1  
    movlw 0x3f  
    movwf TRISA  
    clrf TRISC  
    BCF STATUS, RP0  
    clrf PORTC
```

```
LOOPMAIN:  
    BTFSC PORTA, 2  
    goto izquierda  
    BTFSC PORTA, 1  
    goto centro  
    BTFSC PORTA, 0  
    goto derecha  
    goto LOOPMAIN
```

```
izquierda:  
    MOVLW 0x01  
    MOVWF PORTC  
    ; retardo lms  
    movlw 0x03  
    movwf regini  
    movlw 0xFF  
    movwf regini2  
    call RETARDO  
    ;call RETARDO  
    clrf PORTC  
    ; retardo 20 ms  
    movlw 0x82  
    movwf regini  
    movlw 0xFF  
    movwf regini2  
    call RETARDO  
    goto LOOPMAIN
```

```
centro:  
    MOVLW 0x01  
    MOVWF PORTC  
    ; retardo lms  
    movlw 0x09  
    movwf regini  
    movlw 0xFF  
    movwf regini2  
    call RETARDO  
    ;call RETARDO  
    clrf PORTC  
    ; retardo 20 ms  
    movlw 0x82  
    movwf regini  
    movlw 0xFF  
    movwf regini2  
    call RETARDO  
    goto LOOPMAIN
```

```
derecha:  
    MOVLW 0x01  
    MOVWF PORTC  
    ; retardo lms  
    movlw 0x10  
    movwf regini  
    movlw 0xFF  
    movwf regini2  
    call RETARDO  
    ;call RETARDO  
    clrf PORTC  
    ; retardo 20 ms  
    movlw 0x82  
    movwf regini  
    movlw 0xFF  
    movwf regini2  
    call RETARDO  
    goto LOOPMAIN
```

```
RETARDO:  
    movf regini, w  
    movwf reg1  
LOOPPRET1:  
    MOVF regini2, w  
    MOVWF reg2  
LOOPPRET2:  
    DECFSZ reg2  
    goto LOOPPRET1  
    DECFSZ reg1  
    goto LOOPPRET2  
return
```

end



CONCLUSIÓN

- Sanjuan Aldape Diana Paola

Para realizar esta práctica se requería conocer el funcionamiento de los puertos paralelos de entrada y salida. Para el motor de corriente directa y motor a pasos, se usó el puerto A como puerto de entrada, el puerto B en bajo se usó para el motor de corriente directa y en alto para el motor a pasos. También se usó un retardo de 20 ms para el motor a pasos y para el servo. De esta manera se logró que los motores se movieran en el sentido solicitado cuando se les indicaba mediante los switches.

- Esparza Fuentes Jorge Luis

En esta práctica se logró desarrollar soluciones a los problemas propuestos con base en el control de los puertos paralelos a partir del datasheet del controlador se diseñaron las soluciones para controlar los motores de corriente directa, servomotores y motores a paso, utilizando puentes H o drivers L293D, de esta forma se obtuvieron los resultados esperados y de forma óptima.