

# ***Control de motores PaP unipolares mediante PIC16F84***

Por: Alejandro Alonso

Fecha: 29/12/2002

---

## **Objetivo**

El proyecto consiste en un montaje hardware y un programa en ensamblador para controlar la velocidad de un motor paso a paso unipolar.

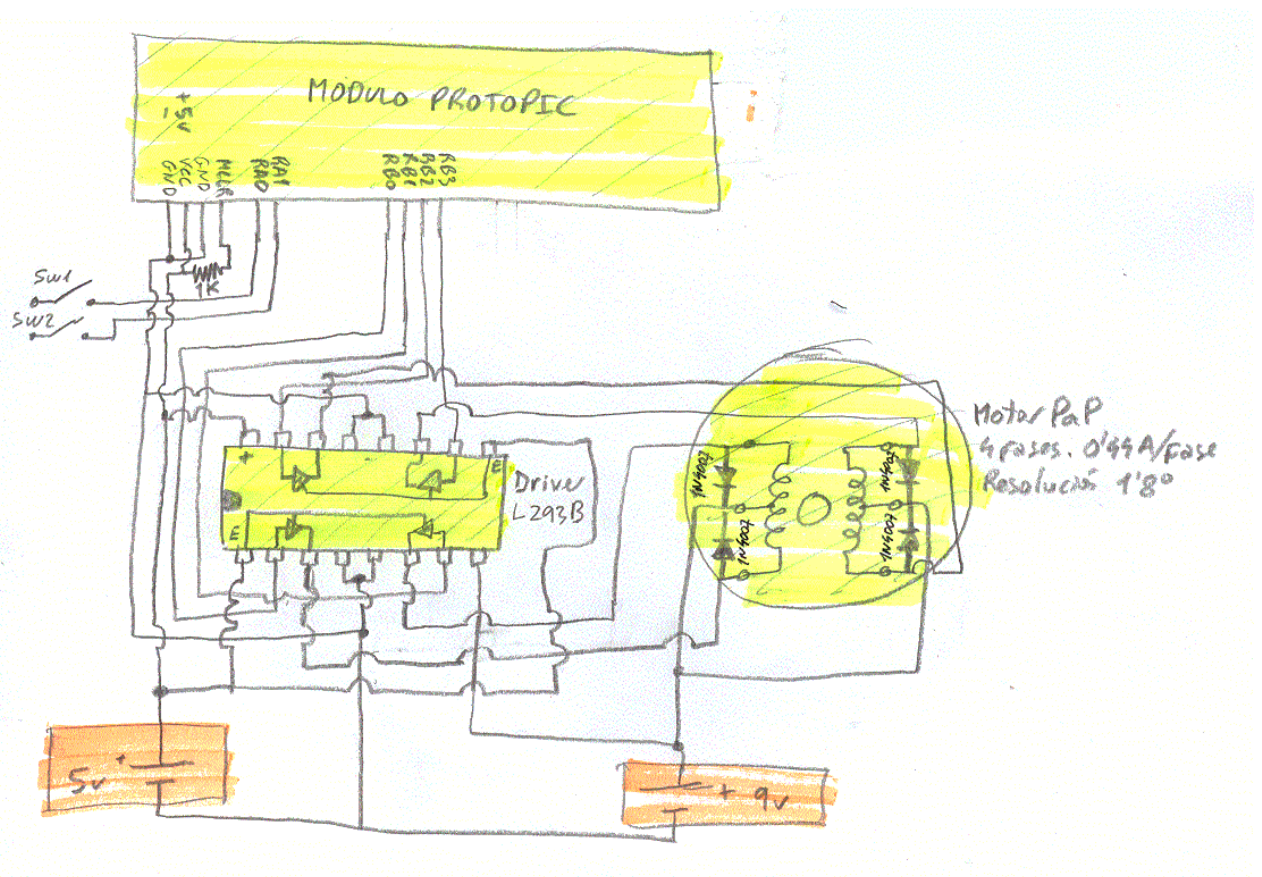
Para indicar al procesador la velocidad a la que se quiere que funcione el motor, existen dos switches cuya combinación de posiciones permite establecer cuatro velocidades diferentes. Según la velocidad se enviarán los pulsos de la secuencia paso a paso a mayor o menor frecuencia.

En este montaje no existe bucle de realimentación. Es decir que el procesador se limita a enviar la secuencia de pulsos al driver del motor, sin verificar que la velocidad alcanzada sea la deseada.

## El montaje hardware

A continuación se muestra plano manual del montaje hardware. Básicamente está formado por 3 bloques:

- El módulo protopic, que no es más que un grabador de PICs.
- El motor PaP
- El driver del motor

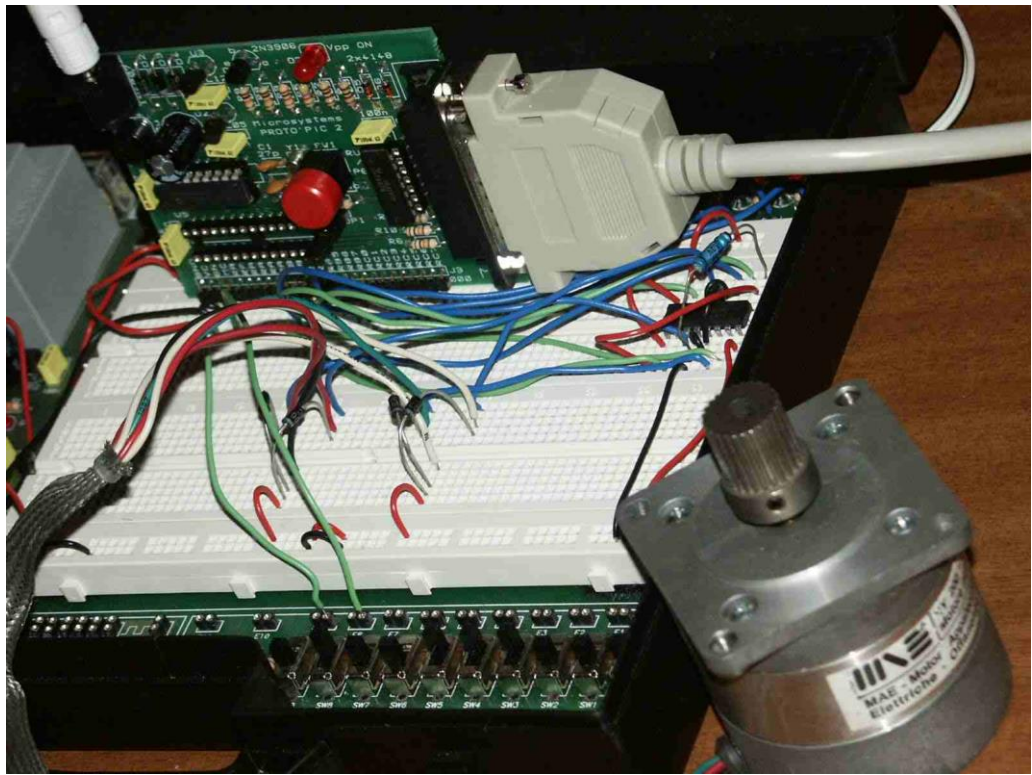


La alimentación de la lógica y del motor está separada no solo para evitar interferencias, sino también para poder manejar diferentes voltajes. Así el motor está alimentado por otra fuente de 9v independiente de la de 5V utilizada por la lógica TTL. Dicha corriente llega al motor a través del driver L293B

El motor tiene cuatro diodos a la entrada de sus terminales para disipar la corriente de retorno producida al parar el motor. Esta corriente es producida debido a que el motor actúa como una bobina autoinducida.

La velocidad máxima que hemos alcanzado con el motor es de 73 rpm

El montaje queda como se muestra en la foto:



## El Programa

El programa se ha realizado en ensamblador para el PIC 16F84.

Simplemente lee la combinación de los switches 1 y 2. Según dicha combinación cargará en la variable “Velocidad” un valor diferente. Al contrario de lo que se debiera esperar, cuanto mayor sea el valor de velocidad, menor será la velocidad del motor y viceversa. Esta variable en realidad servirá para hacer una pausa entre cada combinación de valores enviados al driver del motor. La combinación de valores enviados a RB0, RB1, RB2 y RB3 que son los que impulsarán al motor a rotar son los siguiente:

RB0, RB1, RB2, RB3
0111
1101
1011
1110

La combinación es así porque los terminales centrales de las bobinas del motor están conectados a positivo, con lo que cada bobina se activa por nivel bajo.

El código del programa es el siguiente:

```
; Motor03
; Programa para control de motores paso a paso unipolares
; Por: Alejandro Alonso
; Fecha: 29/12/2002
; Función:
;
; Simplemente hace rotar el motor

list      p=16f84
include   "P16F84a.INC"

Contador   EQU    0x0C    ;Contador multiuso
Velocidad  EQU    0x0F    ;Inverso de velocidad

; Definiciones bits del registro RA

Switch1    EQU    0       ;RA0 - Switches que definen la...
Switch2    EQU    1       ;RA1 - velocidad del motor
;Atención, RA4 en modo salida trabaja en colector abierto

org    0
goto   INICIO
org    5

; -----
INICIO          ;Inicio del cuerpo del programa

bsf     STATUS,RP0          ;Apunta a banco 1
movlw   b'11111111'         ;Establece puerta A como ENTRADA
movwf   TRISA               ;
movlw   b'00000000'         ;Establece puerta B como SALIDA
```

```

movwf TRISB          ;
movlw b'10000011'    ;Configuracion OPTION para TMR0
movwf OPTION_REG
bcf  STATUS,RP0      ;Apunta a banco 0

movlw b'00000000'    ;Establece interrupciones
movwf INTCON          ;anuladas

clrf  PORTA

```

**BUCLE** ;Bucle principal del programa

```

;Chequeamos switches para asignar velocidad

btfss PORTA,Switch1   ;Vemos valor Switch1
goto  Sw_0x           ;Sw1=0
btfss PORTA,Switch2   ;Sw1=1. Vemos valor Switch2
goto  Sw_10           ;Sw1=1, Sw2=0
goto  Sw_11           ;Sw1=1, Sw2=1
Sw_0x btfss PORTA,Switch2 ;Sw1=0. Vemos valor Switch2
goto  Sw_00           ;Sw1=0, Sw2=0
goto  Sw_01           ;Sw1=0, Sw2=1

Sw_00 movlw 16
goto  Sw_Ok
Sw_01 movlw 8
goto  Sw_Ok
Sw_10 movlw 2
goto  Sw_Ok
Sw_11 movlw 1
goto  Sw_Ok

Sw_Ok movwf Velocidad ;Cargamos el valor seleccionado en Velocidad

Ciclo movlw b'00001110' ; un paso del motor
movwf PORTB
Movf  Velocidad,W      ; valor inverso de velocidad...
movwf Contador         ;...se carga en "contador"
call  Retardo

movlw b'00001011'     ; un paso del motor
movwf PORTB
Movf  Velocidad,W      ; valor inverso de velocidad...
movwf Contador         ;...se carga en "contador"
call  Retardo

movlw b'00001101'     ; un paso del motor
movwf PORTB
Movf  Velocidad,W      ; valor inverso de velocidad...
movwf Contador         ;...se carga en "contador"
call  Retardo

movlw b'00000111'     ; un paso del motor
movwf PORTB
Movf  Velocidad,W      ; valor inverso de velocidad...
movwf Contador         ;...se carga en "contador"
call  Retardo

GOTO  BUCLE

```

; -----

; Subrutinas

```
Retardo           ;Provoca un retardo según el valor de "Contador"
Bucle1      movlw   00                ;Inicializacion bucle interno
            movwf   TMR0
Bucle2      btfss   INTCON,T0IF       ;¿Terminado bucle interno?
            goto    Bucle2           ;No, continuar bucle interno
            bcf     INTCON,T0IF       ;Si, bajar bandera
            decfsz  Contador,F        ;Decrementar contador bucle externo
            goto    Bucle1           ;y repetir bucle externo hasta fin
            RETURN
```

```
Fin
      END
```