
C mo Programar un PIC

Introducción

En la actualidad es muy fácil construir productos electrónicos con bajo costo, espacio reducido y con características multifuncionales; de hecho, son muchos los circuitos decodificadores de señales de TV contruidos con pequeños controladores programables integrados que contienen programas sencillos. En este capítulo, explicaremos cómo se programa un PIC con la ayuda de una computadora, sin que para ello se necesiten complicados circuitos adicionales. También daremos las direcciones de Internet para recabar información sobre la forma de "leer" el programa contenido en un PIC y la forma de obtener prestaciones especiales.

En la creación y diseño de un nuevo producto electrónico, la tendencia actual es tratar de respetar los términos descriptivos "*bajos costos*", "*ultra-compacto*" y "*multifuncional*". Lograr un producto con todas estas características era sumamente difícil. Afortunadamente, en los últimos años, diseñar un producto con esos atributos se ha tornado más fácil gracias al desarrollo de dispositivos programables como la familia de microcontroladores PIC de Tecnologías Microchip.

Estructura de un Programador de PICs

En la actualidad es muy fácil construir productos electrónicos con bajo costo, espacio reducido y con características multi-

funcionales; de hecho, son muchos los circuitos decodificadores de señales de TV contruidos con pequeños controladores programables integrados que contienen programas sencillos. Nuestro objetivo es que pueda programar un PIC con la ayuda de una computadora, sin que para ello se necesiten complicados circuitos adicionales. Se necesita un dispositivo de programación para incorporar el software de los chips que venimos describiendo desde hace varios números (nos referimos a los PICs). El costo de un dispositivo de estas características suele ser muy alto, razón por la cual el lector le "huye" al diseño de sistemas electrónicos con estos componentes. Obviamente, el programador PIC presentado en este libro necesita algunos componentes accesorios para ser construido pero resulta una solución ideal para quienes poseen computadora o puedan emplear la de un amigo o técnico conocido. El rasgo más importante de esta unidad es que no necesita partes especializadas y es un circuito muy simple.

Luego de armado se lo conecta al port (puerto) de la impresora de cualquier PC (el puerto no necesita ser bi-direccional), y ejecutar un software (programa) para iniciar la programación de los microcontroladores.

Haremos referencia a la serie PIC 16F84, 16F83 y 16C84, por ser componentes de bajo costo que se consiguen con facilidad .

La información que aquí damos fue re-

dactada a partir del "website" de Microchip Inc., empresa fabricante de estos componentes.

Cabe aclarar que en las páginas de Internet de Microchip Inc. puede encontrar toda la información necesaria para el diseño y construcción de circuitos a partir de microcontroladores PIC de la serie 16XXX.

Tal como diimos en el capítulo 1, un PIC, como la mayoría de los microcontroladores, es una computadora pequeña con CPU, ROM, RAM y circuitos I/O (de entrada y salida) en un solo chip.

Los Microcontroladores constituyen el segmento de mayor crecimiento en la industria electrónica (existen PICs en decodificadores de señales de TV por cable, relojes, microondas, equipos de música, lavapropas, etc.).

Se puede considerar el microcontrolador como un circuito integrado que puede ser *"personalizado"* si se escribe un programa en lenguaje de máquina (assembly). Dicho de otra manera, cargando el programa en el área ROM del microcontrolador, obtendremos un circuito integrado específico para nuestras necesidades. Algunas veces el programa está diseñado para ser algo más que un oscilador o una llave lógica, pero lo importante es que el PIC realizará exactamente lo que usted le indique.

La familia PIC de la empresa Microchip Inc. es muy simple de operar, es de bajo costo y gran parte del software utilizado para crear el código de programa necesario para la programación del PIC está disponible en el website de Microchip, cuya dirección es la siguiente:

www.microchip.com

En la figura 1 damos una muestra de la pantalla que hemos obtenido

al navegar por Internet. De los tipos de PICs disponibles, los más fáciles para trabajar, según el resultado de mi búsqueda (vea la figura 2) son el 16F83, el 16F84 y el 16C84. Estos modelos particulares son los que soporta nuestro programador.

Cuestan menos de \$10 cada uno, y su ROM es eléctricamente borrrable, de modo que no necesita una luz ultravioleta para borrar y reprogramar los dispositivos.

El 16F84 es el más empleado, tiene 68 bytes de RAM y 1.024 palabras de memoria de programa.

La memoria de programa es un EE-PROM tipo flash, que puede ser reescrito más de un millón de veces. Cualquier información almacenada será preservada por más de cuarenta años, sin alimentación alguna o señal de refresco.

El 16C84 es muy similar, pero utiliza un tipo de EEPROM más viejo.

El 16F83 puede ser considerado *"el hermano menor"*, ya que tiene sólo la mitad de memoria; aunque tiene a su favor que es el de costo más bajo costo.

Estos PICs no requieren cristales de cuarzo o resonadores para los circuitos de reloj. Emplea un resistor o un capacitador para fijar la frecuencia de oscilación.



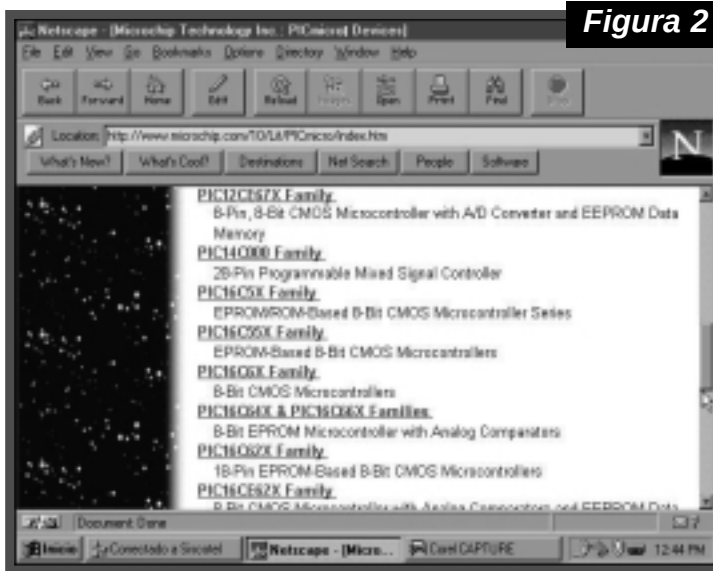


Figura 2

tor 13 y en el momento de ser grabado, se eleva la tensión de la pata 12 a 5V durante al menos $0,1\mu s$ antes de ser bajado de nuevo a nivel tierra.

Los datos que se dirigen al chip contienen comandos que especifican los numerosos pasos del proceso de programación y los datos que serán guardados en la ROM del chip.

Para verificar que el PIC ha sido programado correctamente, compruebe que el PIC también pueda enviar los datos contenidos hacia

el exterior a través de la pata 13.

Para aquellos que estén interesados en conocer los detalles técnicos completos de la programación de los chips PIC, las especificaciones para la programación y su verificación las veremos más adelante.

El diagrama esquemático del Programador para PIC que menciona la empresa fabricante, se muestra en la figura 3. El circuito está diseñado para adaptarse al port de la impresora de cualquier PC.

En la figura 4 mostramos el circuito con el cual hemos realizado varios diseños; em-

La tensión de alimentación para el 16F84 puede ser de 4 a 6V pero existe una versión de este integrado que puede operar inclusive, con 2V.

Tiene 13 pines de entrada/salida, cada uno de los cuales puede ser configurado individualmente para que sea una salida o una entrada

Cómo Programar un PIC de Microchip Inc.

Programar los chips PIC mencionados es muy sencillo:

1º) Se debe aplicar 5V a la pata 14 (con la pata 5 conectada a tierra).

2º) Eleve la tensión de la pata 4 a 12V).

3º) Coloque los datos en la pata 13; dichos datos serán enviados a la ROM de a un bit por vez en concordancia con la señal de reloj. Los datos serán enviados al conec-

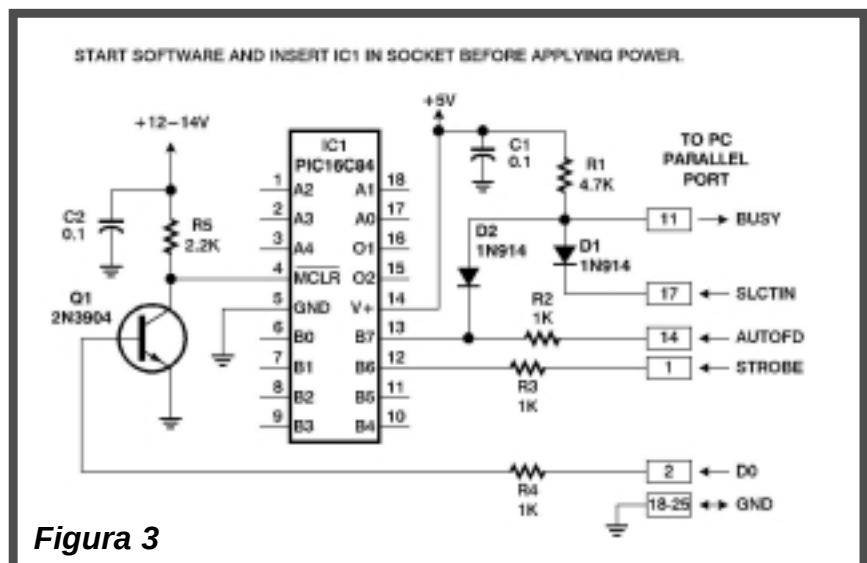


Figura 3

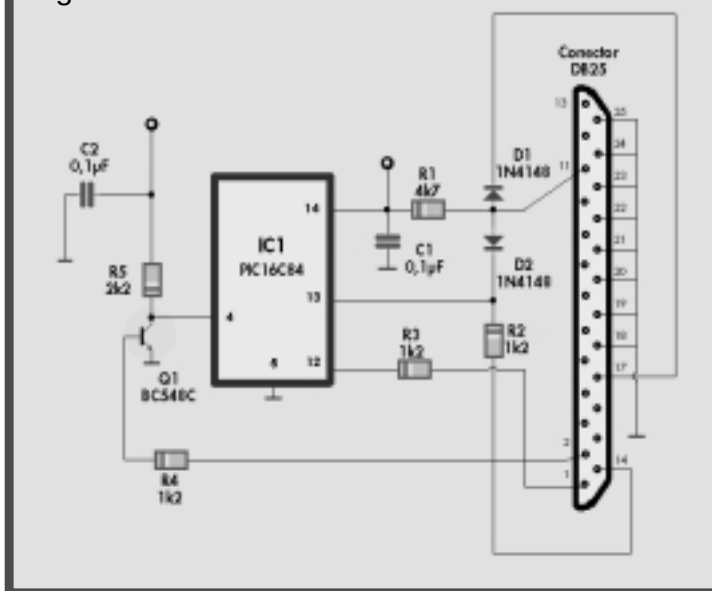
pleamos como computadora una AT 486 con 4MB de RAM y disco rígido de 120MB (también realizamos las pruebas en otra compatible con Pentium 2 de 350MHz y 64MB de RAM).

Los datos de programación y las señales de sincronismo se aplican al PIC a través de R2 y R3. Los diodos D1, D2 y el resistor R1 dejan que el pin 13 del PIC pueda ser utilizado como una entrada para la programación y una salida para la verificación (recuerde que una misma pata de I/O puede ser empleado tanto como entrada o salida). Cuando la pata 17 del conector en el port de la impresora está en estado alto, la PC puede leer los datos desde la pata 13 del PIC a través de la pata 11 del conector de la impresora. De este modo, R1 y D2 se encargan de detener la señal de datos.

Cuando la pata 11 del conector de la impresora está bajo, D1 conduce. El ánodo de D2 es bajado a tierra para bloquear la señal de datos. El chip PIC estará libre para recibir datos desde la pata 14 del port de la impresora. La conexión que crea D1 entre las patas 11 y 17 del conector de la impresora permite que el software de programación detecte si el programador está conectado al port de la impresora.

Los capacitores C1 y C2 eliminan el ruido que pueden producir los cables que llevan la corriente DC. La tensión necesaria para que se produzca la programación es comandada por Q1 a través del pin 2 del port

Figura 4



de la impresora. Se requieren dos suministros de tensión para poner en marcha el Programador PIC; 5V para la pata 14 y 12V para la pata 4.

Estas tensiones pueden ser tomadas desde la misma PC o desde una fuente externa. La tensión a aplicar en la pata 4 debe estar comprendida entre 12V y 14V. Asegúrese de que la línea de 12V tenga realmente 12V como mínimo. Si desea emplear fuentes externas puede utilizar el esquema de suministro de tensión mostrado en la figura 5.

El circuito del programador es tan simple que puede ser construido fácilmente en una pequeña placa de circuito impreso con uso de técnicas de construcción estandarizadas. El montaje por separado del conector del cable hace que el Programador PIC sea más fácil de utilizar, simplemente se lo conecta al port de la impresora de la PC con un multicable de 30 cm de largo.

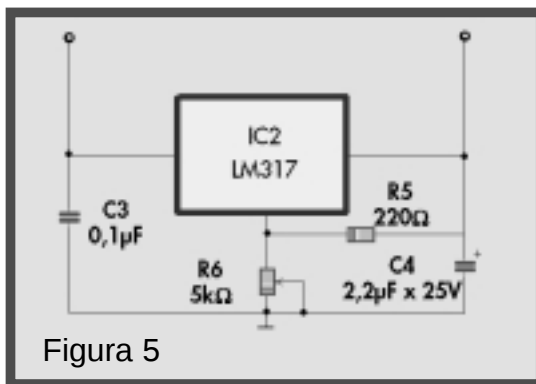


Figura 5

Programación del PIC

La manera más fácil de aprender a utilizar el Programador PIC es escribir un programa simple e incorporarlo a un chip. Este modo de aprender las características del programador, también le brindará la oportunidad de verificar su hardware y software.

Como ejemplo, veamos cómo programar un circuito que enciende un LED después de otro, en secuencias sucesivas. Este circuito podría hacerlo con un oscilador y un CD4017.

Si embargo, también podemos hacer el circuito con un PIC, tal como se muestra en la figura 6.

Si bien este circuito puede costar más caro que un circuito digital tradicional, es más pequeño y no necesita tanto espacio físico para contener todos sus componentes. De todos modos, hay otro argumento para utilizar el PIC en una tarea de estas características: *su inteligencia*. Se pue-

den generar diferentes modelos o seleccionar múltiples modelos con la adición de un simple mecanismo interruptor. Con componentes discretos, el circuito debiera ser descartado y rediseñado para adaptarse a este tipo de cambios. Con un PIC, sólo es cuestión de modificar el programa, borrar el PIC y reprogramarlo. Para programar un PIC desde una PC hace falta un programa y conocer el set de instrucciones de microcontrolador, pero no se asuste, esto no es difícil.

Para hacer que nuestro controlador prenda leds en secuencia, deberá programar que todas las patas **I/O** sean de salida y hacer que cada una de estas patas tenga un estado alto en forma secuencial, con un período que podemos fijar nosotros.

De esta manera, el programa, escrito en lenguaje **Assembly PIC**, se muestra en la Tabla 1. La parte más importante de cualquier programa es la documentación. Si deja notas en el programa, recordará

qué es lo que estaba intentando hacer cuando revise el programa, la vez siguiente. En el programa, estas notas aparecen en cualquier renglón que se inicie con punto y coma. Estos renglones serán ignorados por el software, mediante un montador que convierte el texto del programa en números binarios que el PIC reconoce como instrucciones.

Las primeras tres instrucciones son:

```
processor, 16f84
include <p16f84.inc>L
_config,
```

Estas son instrucciones pa-

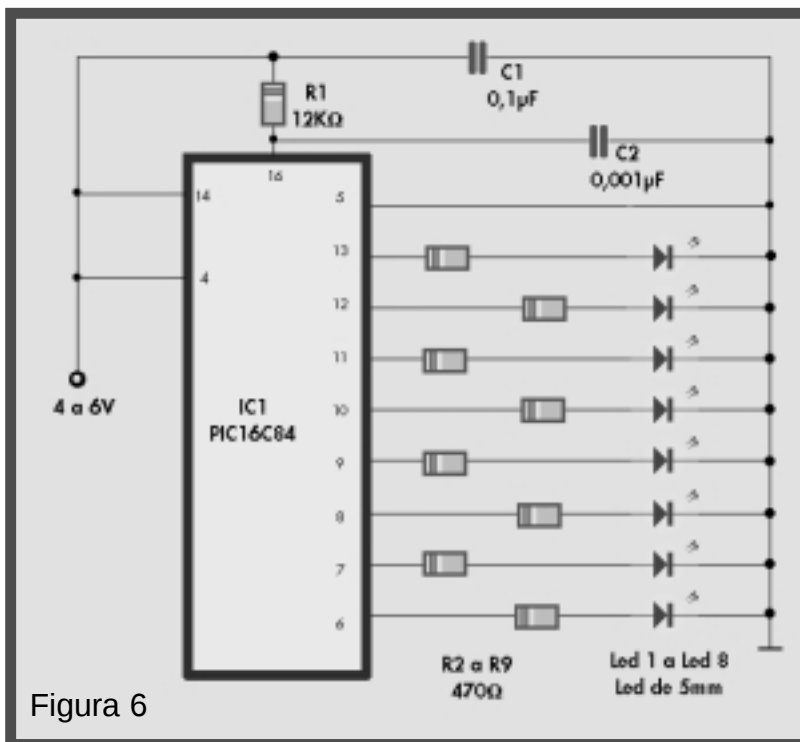


Figura 6

ra el técnico. La primera instrucción le indica al técnico que utilice las instrucciones del 16F84. La segunda le indica que debe incluir un conjunto de constantes predeterminadas en un archivo llamado p16F84.INC. Finalmente, la tercera establece varios bits de configuración en el PIC para activar o desactivar determinadas características del hardware.

En este ejemplo, el chip utilizará su oscilador RC, su WDT (cronómetro "vigilante") será desactivado y el PWRTE (reloj automático de reseteado) será activado. De este modo, el PIC "rebooteará" (se pondrá en marcha) con cualquier tensión que se le aplique, pero no "rebooteará" automáticamente muchas veces por segundo.

Es importante darle uso a la instrucción **_config** en todos los programas utilizados con el Programador PIC. El programa del técnico no realizará la programación real, sólo creará un archivo con los números que serán transferidos al chip PIC como un segundo paso.

Las dos instrucciones que reservan espacio de memoria en el RAM de los PICs para dos variables, que se llamarán "J" y "K". El proceso es similar a estipular variables en **BASIC**, con la diferencia de que se deben determinar las ubicaciones físicas de RAM que se utilizarán. En este caso, las ubicaciones (en una numeración hexadecimal) son 1E y 1F. Estas ubicaciones serán utilizadas para almacenar calculadores que registren la cantidad de veces que una subrutina (loop) ha sido repetida.

La instrucción **org** le indica al montador que el programa se inicia en la ubicación 0 de la memoria de programa y que el programa real será el próximo.

La primera instrucción real del PIC es una instrucción **movlw** que borra el contenido del registro W. Ese número es luego

copiado en el registrador de control TRIS para el Port B y establece que las patas 6 a13 (que son manejadas por el Port B) sean terminales de salida en vez de serlo

```
;Código Assembly para microcontrolador PIC16F84 como
;circuito secuencial.

;Encendido de Leds en secuencia rotativa.

;Con una frecuencia de 75kHz, cada led permanecerá
;encendido durante 0,5segundos.

; Configuración de la CPU
; (se trata de un 16F84 con oscilador RC
; y grabado con tensión positiva durante
; el pulso en estado alto)

    procesor 16f84
    include  <p16f84.inc>
    _config _RC_osc & _WDT_OFF & _PWRTE_ON

; serán variables las dos direcciones de memoria

J    equ    H'1F'    ; J = dirección hexa 1F
K    equ    H'1E'    ; K = dirección hexa 1E

;Programa

    org     0        ; comienzo en la dirección 0

; Elijo al Port B como salida y la inicializo

    movlw   B'00000000'    ; w:= 00000000 binario
    tris    PORTB          ;portB ctrl register := w
    movlw   B'00000001'    ; w:= 00000001 binario
    movwf   PORTB          ; port B itself := w

; Rote el bit del port B hacia la izquierda

    mloop:   rif           PORTB,f

; Espere un tiempo para ejecutar nuevamente la subrutina o
loop
    movlw   d'50'          ; w = 50 en decimal
    movwf   J              ; J := w
jloop:      movwf   K        ; K := w
kloop      decfsz   kKf      ;K = K-1, salte si K = 0
           goto     kloop
           decfsz   J,f      ;J = J-1, salte si J = 0
           goto     jloop

;lo hacemos nuevamente
           goto     mloop

end
```

Tabla 1

de entrada. Si hubiese querido que todas fuesen entradas debiera haber copiado en el port B la palabra "11111111"

Volviendo a nuestro programa, seguidamente se introduce el número binario 00000001 en el registrador W y lo copia al Port B. Esta acción enciende el LED conectado a la pata 6, dado que ahora, el terminal Pata 0 pin 6, que es de salida, tiene un nivel "1", es decir, un nivel alto. Pero antes de que tenga tiempo de visualizar la llegada del pulso al LED, el programa ejecutará una instrucción *rlf* que rota los contenidos del Port B a la izquierda y cambia los datos a 000000-10. Esta acción encenderá el segundo LED conectado al conector 7. La repetición de la instrucción producirá 00000100, luego 00001000, y así sucesivamente, esto hará que los diferentes leds se enciendan en secuencia.

Entre rotaciones, el programa necesita esperar 1/2 segundo aproximadamente, de modo que la acción no es tan rápida como para que no se la pueda apreciar. Esa es la causa por la cual aparece en el programa una subrutina de demora. El programa almacena el número decimal 50 en las ubicaciones de memoria "J" y "K", utilizando la instrucción **decfsz** para la cuenta regresiva de 50 a 0.

Las instrucciones condicionales del PIC son inusuales, y el **descfsz** no es una excepción. Esta instrucción significa: *"Decrementar y omitir la instrucción siguiente si la variable es igual a cero"*.

Normalmente, el resultado no es cero y la parte "ir a" de la instrucción no es omitida. De tal modo, la subrutina (loop) se ejecuta repetidamente hasta que K llega a cero.

El programa real utiliza dos loops (subrutina), una incrustada dentro de la otra.

Finalmente, ir a **mloop** envía la ejecución al comienzo del programa. La instrucción *end* no es una instrucción para el mi-

crocontrolador (CPU); sólo le indica al montador que el programa ha finalizado.

El 16F84 tiene 35 instrucciones CPU diferentes. Como puede ver a partir del programa simple que hemos creado, no debe dominarlas a todas para escribir programas útiles.

Cómo se Construye un decodificador con PIC

Ahora mire que fácil sería construir una llave electrónica con PIC que se active luego de un cierto tiempo, o cuando se recibe alguna señal. Habría que programar una pata I/O como entrada y otra como salida, luego el programa debiera reconocer una señal presente en la pata fijada como entrada y luego de un tiempo activar la salida que se encargaría de poner en marcha un circuito. *¿Le suena conocido?*, esto es lo que hacen normalmente los decodificadores de las señales de TV por cable.

Por lo tanto, si sabe cómo es una señal codificada y obtiene el set de instrucciones del PIC de Internet, no tendrá inconvenientes en construir un programa para su PIC, con el objeto de decodificar señales de TV.

Seguramente Ud. se preguntará, *¿Por qué no doy el programa en este libro?*, muy sencillo, considero que debo darle las herramientas para que aprenda y no inducirlo a que *"arme decodificadores con fines comerciales"*.

Cómo obtener un Programa para "programar" nuestro PIC

En el Website de Microchip podrá encontrar hojas de datos, notas de aplica-

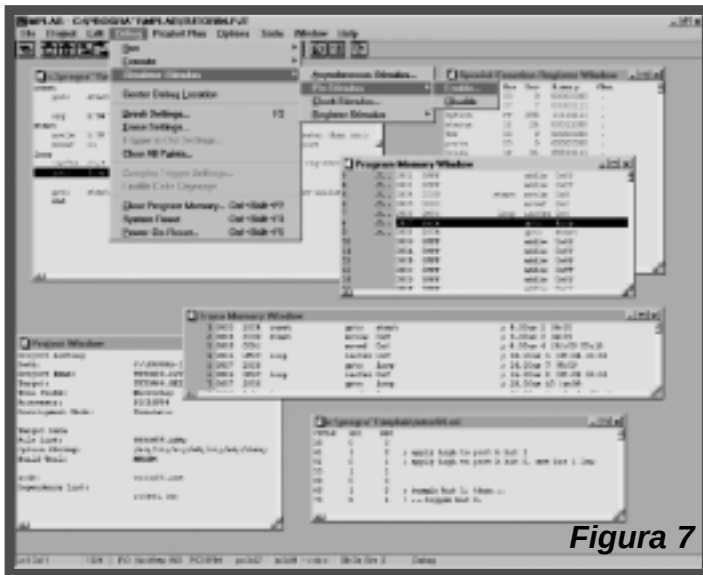


Figura 7

ción y, lo mejor de todo, el MPLAB, un programa de desarrollo para simular y verificar programas PIC. El mismo está diseñado para ser ejecutado bajo Windows.

Particularmente, obtuve el programa de la dirección:

<http://www.microchip.com/10/Tools/MPLAB/index.htm>

Cuando acceda a esta dirección, verá que el programa precisa de 6 disquetes y Ud. debe bajar cada uno de ellos a su computadora.; luego deberá descomprimir los archivos y comenzar la instalación (en total pesa unos 8MB).

Una vez hecho esto el mismo programa le da toda la información necesaria para conseguir su propósito (en el capítulo 3 detallamos el uso de este programa). Una muestra de su pantalla, se ve en la figura 7.

No debe asustarse si nunca manejó una PC, el programa le indica todo lo que debe hacer; y lo mejor, es GRATIS.

El software MPLAB le permitirá editar programas en lenguaje Assembly (también

llamados códigos de fuente), compilarlos en un código de objeto y luego analizar el código binario resultante para ver qué acción realizará el microcontrolador. De este modo, podrá localizar los errores lógicos en su programación antes de ejecutar cualquier código en el hardware.

El uso del software de Microchip está muy bien documentado en las páginas de Internet.

Prepárese para recibir mensajes de error cuando compile el programa. El **MPASM** se quejará moderadamente de que no utilizará la ins-

trucción **tris**. Microchip le ha restado importancia a esta instrucción y algunos procesadores PIC del futuro podrían no registrarla. Desde la perspectiva del software, uno de los mejores rasgos del PIC es su llamada compatibilidad de código de fuente. Si su diseño supera los recursos de chips con los cuales inició el programa, podrá utilizar otro chip con mayores recursos sin necesidad de reescribir el programa desde el principio. Para nuestros propósitos, la utilización de la instrucción **tris** en el 16C84, 16F84 y 16F83 trabaja de modo aceptable. Además, la vía alternativa de configurar el Puerto B para la salida es mucho más complicada.

Otro programa más sencillo para "programar" nuestro PIC se denomina **NOPPP** y puede obtenerlo de la dirección:

<ftp://ftp.gernsback.com/pub/EN/noppp.zip>.

del Website de Gernsback. Este programa MS-DOS se ejecuta bajo Windows 3.x o Windows 95. De cualquier modo, si ejecuta el programa bajo Windows 3.1, trabajará mejor si lo ejecuta bajo la aplicación de "pantalla completa" que si lo hace en

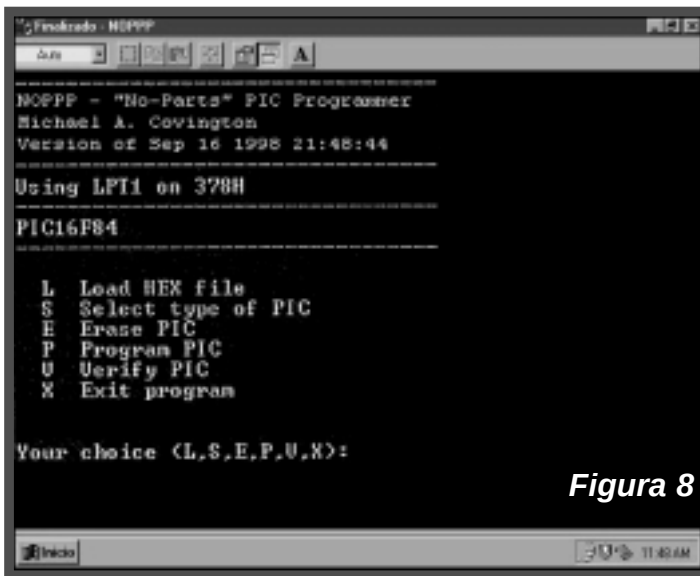


Figura 8

una ventana (veremos cómo programar con esta aplicación en el capítulo 6).

El timing es esencial para los pulsos de programación y las aplicaciones DOS de pantalla completa toman el control total de la computadora.

Si por alguna razón tiene dificultades en ejecutar el programa NOPPP bajo Windows 3.1, intente salir a un entorno DOS y ejecútelo desde allí. También puede ejecutar el programa bajo **OS/2**; si lo hace, asegúrese de configurar el **HW_TIMER** en "on" en las configuraciones DOS para el programa.

Para hacer la programación, el primer paso es conectar el Programador PIC al port de la impresora de la PC e iniciar el programa NOPPP sin ninguna tensión conectada al programador. Si la línea de 5V está conectada a tierra, el software no podrá detectar el diodo D1, y asumirá que el programador no está conectado al port de la impresora.

Si todo se encuentra en orden, debería ver una pantalla similar a la que se muestra en la figura 8. El menú de opciones es auto-explicativo. En general, debería cargar un archivo de código de objeto (con una extensión **.HEX** en el nombre de archi-

vo) en la memoria, seleccionar el tipo de PIC que programará, programar el software y luego verificar que el código fue correctamente programado en el chip. También podrá borrar un PIC que ya ha sido programado para reutilizar o actualizar la programación.

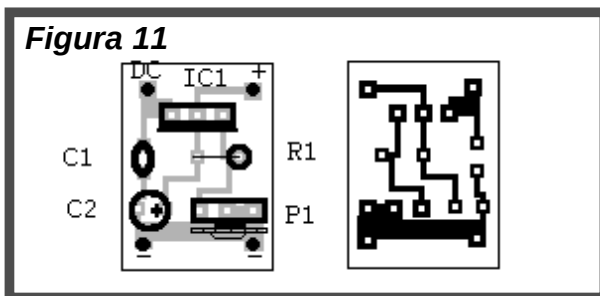
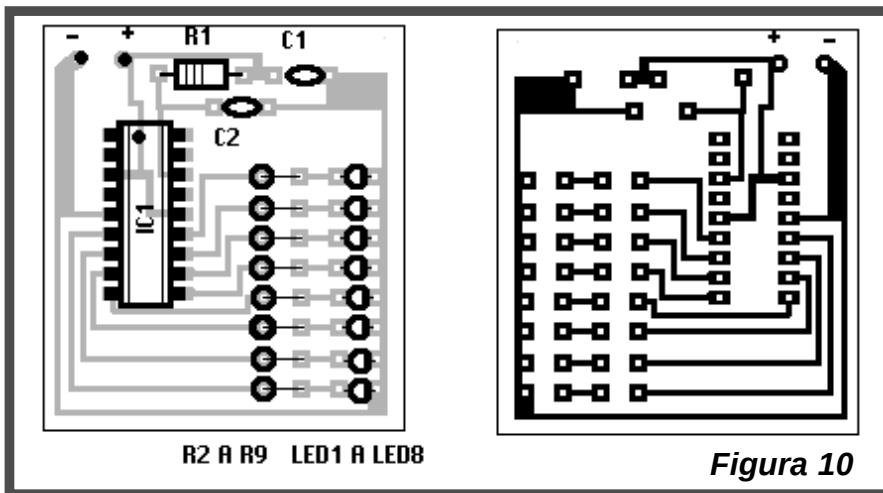
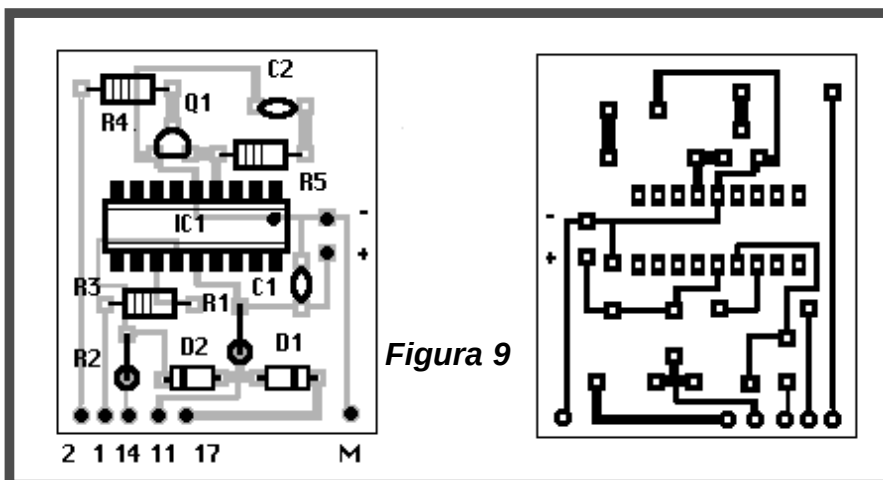
Una advertencia:

Nunca inserte o remueva un PIC de un programador mientras la tensión enviada al programador se encuentra activada. Cuando programe un PIC, el software le indicará qué acciones debe ejecutar y cuándo realizarlas. Dado que el software del programador requiere un timing engañoso, fue escrito para ejecutarse como un programa DOS.

Recuerde que los pulsos de medición para la programación del PIC tienen que durar por lo menos 0,1µs. En la práctica, son lo suficientemente largos como para superar cualquier señal de "rebotes" en los cables.

De todos modos, no deben ser demasiado largos, para que no tornen extremadamente lento el proceso de programación. También es importante que el tiempo del pulso no dependa de la velocidad de la CPU de la computadora. El software fue escrito especialmente para ser ejecutado en cualquier IBM compatible, desde una XT a las últimas Pentium.

Cabe aclarar una vez más que para programar o leer un PIC se debe tener la correspondiente hoja de datos, para saber en qué pata se introducen los datos y cómo se realiza el proceso de programación, también es necesario conocer el set de instrucciones del microcontrolador y saber "*pasar al papel*" las ideas que uno tenga en la mente para realizar las diferentes programaciones.



Quienes tengan dificultad para "bajar programas de escrituras de PICs", podrán retirar de esta editorial, contra entrega de un disquete virgen, un programa aplicativo (debe presentar el presente libro).

También podrá bajar aplicaciones de nuestra página web (vea el capítulo 1).

Para terminar, en la figura 9 se da el impreso del programador de la figura 4; en la figura 10 el correspondiente al circuito secuencial y en la 11, el de la fuente de alimentación.

Aclaremos que en sucesivos capítulos daremos ejemplos de programación de PICs, hasta que Ud. se sienta

lo suficientemente “*ducho*” como para realizar sus propios programas.

No damos lista de materiales, dado que en cada figura se detallan los elementos utilizados. *****