

## CARGADORES (PROGRAMADORES) DE PIC, $\mu$ C Y EEPROM

### INTRODUCCIÓN

Para que un PIC funcione como nosotros queremos, es preciso “cargarle” un programa en la “memoria de programa”. El programa debe estar en un lenguaje comprensible por el chip o sea un lenguaje de máquina en código binario, aunque en realidad al PIC debemos introducirle un programa en código hexadecimal e internamente lo convierte a binario para realizar su interpretación.

El programa que debemos cargarle al PIC en código hexadecimal tiene que tener la terminación “hex”. Sin embargo, normalmente se debe escribir “editar” el programa en un lenguaje entendible por el usuario, normalmente conocido como “assembler”. Los PICs basan su programación en un set reducido de instrucciones que permiten editar el programa generando un archivo con extensión “asm”.

Existen programas que se encargan de “traducir” el programa escrito en lenguaje assembler (asm) a programa escrito en lenguaje de máquina (hex); a estos programas se los denomina traductores o ensambladores.

Por lo tanto, para escribir un programa y cargarlo en un PIC precisamos:

*\* Un utilitario o programa que nos permita escribir el juego de sentencias a cargar en la memoria de programa del PIC. Este utilitario es un “editor” y como ejemplo tenemos el MPLAB.*

*\* Un utilitario o programa que nos permita “traducir” el programa escrito en assembler a lenguaje de máquina para que podamos cargarlo en el PIC. A ese programa se lo denomina “ensamblador” y como ejemplo tenemos al MPASM.*

*\* Un utilitario para “cargar” el archivo que tiene el programa en lenguaje de máquina (archivo con extensión hex) en la memoria del PIC. A estos programas se los denomina “cargadores” y como ejemplo tenemos al NOPPP, al PIC PRO y a tantos otros, muchos de los cuales mencionaremos más adelante.*

Pero también existen otros utilitarios que permiten “simular” y “emular” un programa para ver cómo se comporta antes de cargarlo en el PIC.

El simulador permite detectar errores en el programa (el MPLAB permite simular) para poder corregirlos. El emulador permite que “una computadora se comporte como PIC” siendo las patas del puerto, las correspondientes a las del PIC. Esto quiere decir que si queremos hacer circuitos con PICs, para poder predisponerlo adecuadamente precisaremos:

*\* Programa MPLAB: Para editar (escribir) un programa en lenguaje assembler y así generar el archivo con extensión asm. El MPLAB también me permite “simular” el funcionamiento del programa para detectar errores y corregirlos.*



\* Programa MPASM: Para "ensamblar" o convertir el archivo asm en archivo hex con el objeto de cargarlo en el PIC.

\* Programa NOPPP: O cualquier otro cargador que permita ingresar el archivo con extensión hex en la memoria de programa del PIC. También se puede emplear cualquier otro cargador, hasta incluso se podrían cargar los datos manualmente, tal como veremos oportunamente. Todos los programas cargadores precisarán de un circuito "eléctrico" o hardware para poder realizar la función de carga.

### LOS CARGADORES DE PICs

Dijimos que un cargador de PIC debe poseer, por un lado, el circuito que permita cargar el programa desde una computadora (hardware) y el programa que permita manejar a dicho circuito para realizar la operación de carga (software). El circuito debe ser muy simple.

Vamos a suponer que no tenemos un dispositivo programador a nuestra disposición; es decir que deberemos armar también un dispositivo programador de nuestros PICs que se conecte al puerto paralelo de una PC (el puerto de impresora).

Nuestra intención es que construya un dispositivo para programar PIC que sea muy sencillo, de modo que se pueda armar en el mismo protoboard y que se conectará a la PC por intermedio de una manguera.

Como todos sabemos, un dispositivo para una PC (hardware), siempre necesita un programa que lo controle (software). Nuestra intención es que Ud. no compre nada más que lo imprescindible; así que vamos a usar un software gratuito llamado NOPPP que puede bajarse por Internet.

Puede bajar este programa, el MPLAB, el MPASM y otros programas y utilitarios, siguiendo los pasos dados en la página de contenidos especiales de Saber Electrónica y, lo que es más importante, en forma totalmente GRATUITA.

Para ello diríjase a:

**[www.webelectronica.com.ar](http://www.webelectronica.com.ar)**

Si no tiene Internet o no sabe cómo se hace, lleve este escrito a cualquier cybercafé y pídale al encargado que le baje los programas. Haga doble click sobre el ícono password y luego digite la clave tepec26.

Recuerde que cuando decimos "programar" nos estamos refiriendo a diseñar un programa para un PIC y a la persona que realizó ese trabajo la llamamos "programador"; cuando decimos "cargar" nos estaremos refiriendo a llenar con datos la memoria de un PIC y el dispositivo que realiza esta función lo llamaremos "cargador de PICs". Aunque parezca increíble, nuestro amplio idioma no tiene una palabra precisa que diferenciara a la acción del aparato y se suscitaban graves confusiones al leer, por algo tan simple. En realidad, sí estaba acuñado un término práctico para el dispositivo: "quemador de PICs", pero el mismo sólo se puede aplicar a aquellos PICs que sólo se pueden grabar una sola vez. Por otro lado, el término se puede confundir con un dispositivo que funcionando mal quema a los PICs en lugar de cargarlos, tal como el autor mencionará con respecto a un programador comercial.

Navegando por Internet es posible encontrar gran cantidad de programas cargadores de PICs, nosotros haremos referencia a los cargadores:



- \* Cargador NOPPP
- \* Cargador Quark PRO 2
- \* Cargador PROPIC
- \* Cargador Universal de PICs y EEPROM
- \* Cargador PIC800
- \* Cargador PP84

Obviamente, todos estos circuitos puede “bajarlos libremente” pudiendo tener más información y los enlaces a las páginas web de cada uno de estos cargadores en la página de contenidos especiales que hemos dado anteriormente. Estos cargadores sirven para ser utilizados en diferentes tipos de PICs a saber:

**NOPPP** es un programa de uso libre ampliamente difundido que sirve para cargar PICs del tipo 16C84, 16F83 y 16F84, pero también se dan otros cargadores con sus respectivos programas, que sirven para otros componentes.

**Quark PRO 2** es un programador de los denominados “universales” que permite manejar distintos tipos de microcontroladores (no sólo los PICs de Microchip) y varias memorias EEPROM.

**PICPRO** es un programador avanzado en relación con el NOPPP y sirve para ser montado cuando no se quieren correr riesgos con la PC que se está utilizando para cargar PICs y cuando se desea tener un trabajo libre de errores (el NOPPP es un programa para aprendizaje).

**El Programador Universal** (mal llamado de este modo) sirve para programar: 12C508, 12C509, 12C67X, 16C55X, 16C61, 16C62X, 16C71, 16C71X, 16C8X, 16F8X, 16F873/4/6/7, 16C73/74/76/77. También programa memorias EEPROM de la serie 24LCXX.

**PIC800** es un programador para 16C84, 16F84, 16F873/74/76/77.

**PP84** es un programador sencillo del tipo NOPPP de buenas prestaciones.

### Un Sencillo Entrenador

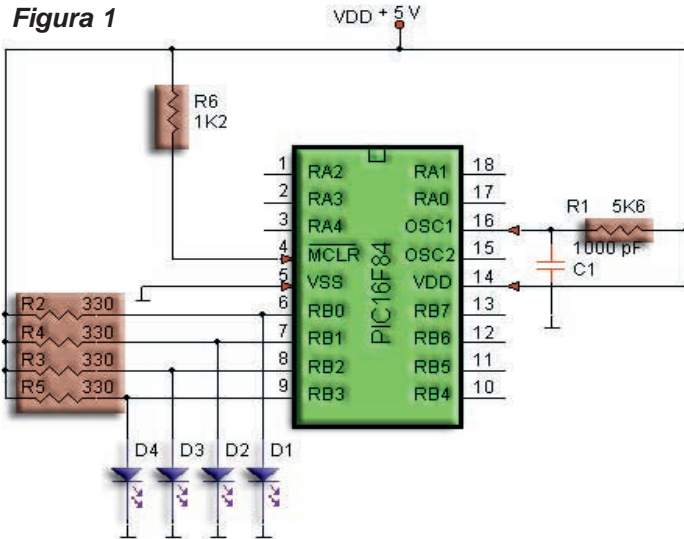
Un circuito con PIC siempre se compone de dos secciones. El circuito del dispositivo deseado y los necesarios componentes periféricos imprescindibles para que el PIC funcione, a saber clock y reset (la primer sección cambia con cada proyecto, en tanto que la segunda puede ser común a muchos proyectos). Ambos circuitos periféricos tienen diferentes opciones: el clock puede ser a cristal cuando se requiera precisión, o a RC (como en nuestro caso) cuando sólo se requiere que la secuencia de encendido tenga un tiempo aproximado. En cuanto al reset, éste puede ser externo con un pulsador o interno y automático, cada vez que se conecta la alimentación de 5V.

Una vez definidas estas cosas se puede dibujar el circuito completo de un sencillo “entrenador” que nos va a permitir verificar si hemos programado y cargado bien a un chip.

Observe primero en la figura 1 la sección periférica imprescindible. El clock está generado por un oscilador RC realizado con el PIC, R1 y C1 (elegidos para obtener una secuencia completa que dure



Figura 1



un segundo aproximadamente), en tanto que el reset se realiza en forma automática, cada vez que se enciende el equipo, por intermedio de R6. Nuestro circuito de aplicación no puede ser más sencillo, sólo utilizamos cuatro LEDs sobre las patas RA0 a RA3 (puerto "A" tomado parcialmente) y sus correspondientes resistores de pull up (resistores a fuente). Nada más simple y fácil de armar en un panel protoboard.

*¿Qué ocurriría si compro los componentes, los armo y pruebo directamente el circuito?*

No va a funcionar porque como el PIC no está instruido para realizar ninguna función y los LEDs van a permanecer todos encendidos. El PIC tiene una memoria interna que debe cargarse con números binarios de 8 dígitos; exactamente tiene 48 posiciones de memoria (numeradas desde el 0 hasta el 2F en números hexadecimales), en donde se le indicará qué tarea debe realizar o qué datos fijos debe cargar para ejecutar esas tareas.

En nuestro caso activar transistores internos en algunas patas del puerto "A" (predispuestas como salidas) de modo de generar cortocircuitos momentáneos a masa en forma secuencial.

El dispositivo con un zócalo para conectar el PIC e instruirlo (cargarle los datos en la memoria de programa) se llama genéricamente "Programador de PICs", pero nosotros en este artículo convenimos en llamarlo "Cargador de PICs" y es como un apéndice de nuestra PC, conectado con un cable al puerto paralelo de la misma.

Si la PC tiene dos puertos paralelos de salida, se usará uno para la impresora y el otro para nuestro programador, si sólo tiene uno, se desconectará provisoriamente la impresora para conectar el programador, o mejor aún, se conectarán ambos dispositivos a través de una caja selectora que se consigue en los negocios de computadoras y se usa para conectar dos impresoras a la misma PC.

Si Ud. tiene un mínimo conocimiento sobre computadoras, sabrá que un dispositivo conectado a la PC es totalmente inútil si no está acompañado de un programa instalado en el disco rígido de la misma. En algunos casos se necesitan dos programas, a saber: el driver del dispositivo y un programa de aplicación que utilice dicho dispositivo. En el caso de dispositivos que se conectan en el puerto paralelo, el programa driver no es necesario porque dicho puerto ya está debidamente habilitado para usar la impresora. Lo que sí se requiere obligatoriamente, es un software de aplicación del programador que suele proveerlo el fabricante del mismo.

Programadores y software de aplicación de los mismos hay muchos. Algunos son muy simples y económicos (tan económicos que muchos se entregan gratuitamente por Internet, es decir que el autor regala el software y da las explicaciones para armar el hardware) y otros son muy complejos y caros. La diferencia entre unos y otros suele ser la posibilidad de aceptar más tipos de PICs (además del 16C84 y 16F84 existen muchos otros) e inclusive microprocesadores o memorias de otras marcas.



También se diferencian en la velocidad a la cual cargan el PIC; los hay de alta productividad que trabajan con un elevado flujo de datos y otros más lentos que sólo sirven para tareas de aprendizaje.

En nuestro caso vamos a trabajar con un software que se llama NOPPP, que es absolutamente gratuito y se puede bajar desde Internet. Este software está previsto para ser usado sólo con los PIC16C84, PIC16C83 y PIC16F84 que son los más utilizados.

Con respecto al hardware, Ud. puede usar el que propone el autor del software que es muy sencillo o usar otro que le propongo yo más adelante y que tiene algunas ventajas con respecto a una mejor forma de los pulsos de programación. El NOPPP es suficientemente sencillo como para implementarlo en un panel protoboard, pero como es un hardware que deberemos usar más adelante para programar otros PICs conviene armarlo en forma más definitiva utilizando una plaqueta de circuito impreso o una plaqueta ojalillada. Si Ud. sólo desea armar el circuito de este artículo puede armar tanto el cargador como el circuito de aplicación en dos lugares distintos del mismo protoboard.

El PIC, como una memoria, tiene una pata que predispone el dispositivo para leer o para escribir. Si la pata 4 del PIC está a un potencial comprendido entre 13 y 14V, el PIC está preparado para escribir los datos que provienen de la PC. Si la pata 4 está por debajo de 6V, el dispositivo está previsto para ser leído. Los datos a leer o escribir se ponen/obtienen de la pata 13 del PIC con la pata 12 del mismo que opera como clock.

Aquellos que conocen el proceso de carga de datos en una memoria serie, no tendrán mayores inconveniente en entender la frase anterior. Para aquellos que no conocen el proceso explicamos lo siguiente:

Un PIC se lee/escribe accediendo a las diferentes posiciones de memoria por la misma pata por la que se obtienen/ingresan los datos (la pata 13). La señal primero elige la posición de memoria a ser leída/escrita, y luego que esa posición está accesible se escriben/leen los datos. El clock que se coloca en la pata 12 sirve para indicar en qué momento se debe transferir la información. Los datos pueden estar sobre la pata 13 todo el tiempo que Ud. desee, ya que no serán ni leídos ni escritos por la PC hasta que se produzca un cambio de estado (de alto para bajo) en la pata 12.

En la figura 2 mostramos las formas de señal indica-

Figura 2

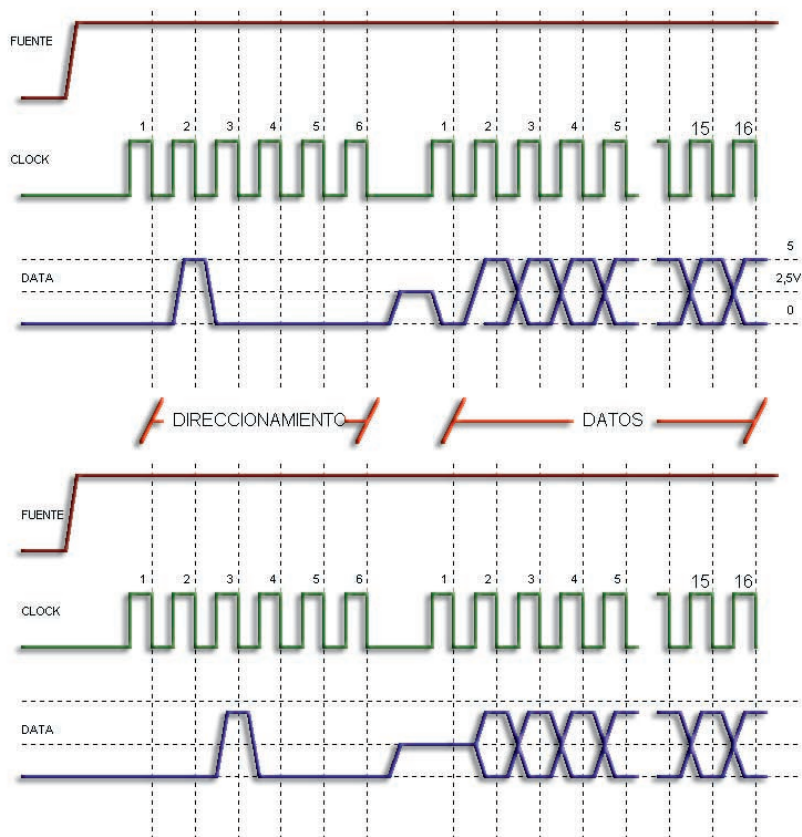
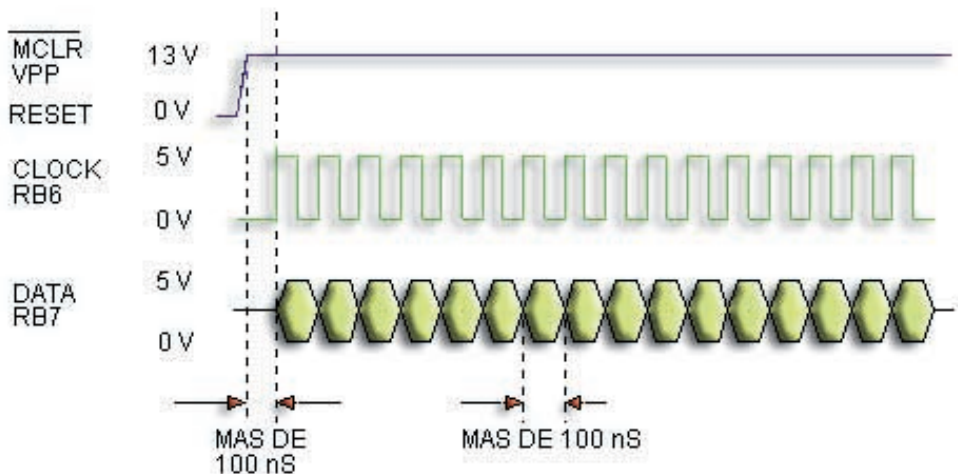
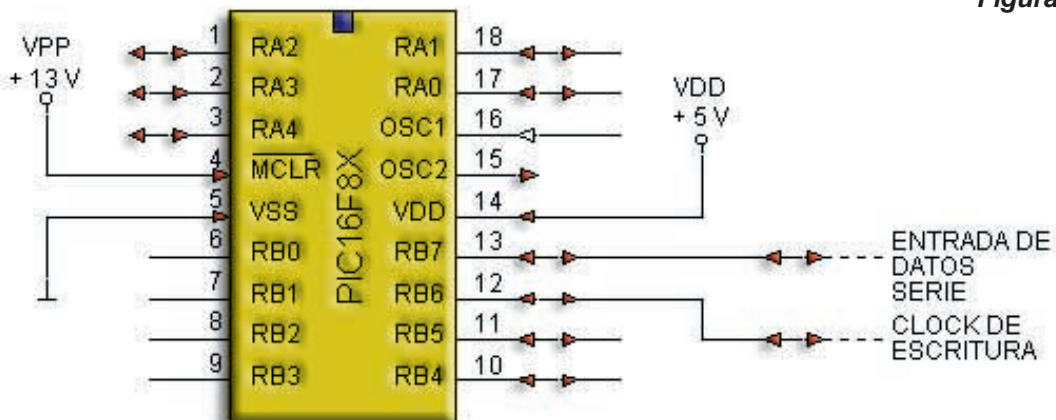




Figura 3



das por Microchip para leer/grabar un PIC. El lector observará que dentro del comando existe una primera sección de 6 bits que indica la posición de memoria a ser leída/escrita y luego la línea de datos permanece en estado de alta impedancia por un intervalo de tiempo que depende de la operación (durante ese tiempo el clock está apagado). Por último ingresan/egresan los datos. Las conexiones de fuente y las señales deben ser aplicadas según una secuencia predeterminada que debe respetarse invariablemente. Los nombres de las señales se aclaran en la figura 3.

A) El primer paso es colocar el PIC en el zócalo (base) del programador con señales y fuentes a potencial de masa.

B) Levantar la tensión de fuente VDD a un potencial de  $5V \pm 0,2V$  por la pata 14 (VDD).

C) Levantar la tensión de fuente VPP a un potencial de  $13V \pm 0,3V$  por la pata 4 (MCLR NEGADO).

D) Esperar en esas condiciones un tiempo superior a 1ms.

E) Posicionar el primer dato en la pata 13 (RB7) con un potencial alto (mayor a 4V) o bajo (menor a 1V).

F) Cuando la pata 12 (RB6) pase a un estado bajo, inferior a 1V, el dato se carga en la memoria.

G) Continuar cargando los datos con el mismo criterio a un ritmo tal que el dato esté presente por lo menos durante 100nS.



H) Cuando todos los datos fueron cargados se debe esperar 1 segundo.

I) Desconectar la fuente de 13V.

J) Desconectar la fuente de 5V.

K) Retirar el micro grabado.

Por sobre todas las cosas debe respetarse el criterio de no sacar un PIC de su zócalo con las fuentes conectadas, porque puede desprogramarse o, peor aún, dañarse definitivamente. También es muy importante respetar las tensiones de fuente y no demorar la llegada de datos, luego conectar VPP (porque podrían cargarse datos falsos por interferencias electromagnéticas).

Como usted puede observar de la descripción anterior, la entrada RB6 es utilizada como clock y la RB7 como entrada de datos en una clásica operación de transferencia de datos en serie.

Observe que hay un tiempo entre el momento en que MCLR NEGADO sale de la condición de reset hasta que aparece el primer dato o el primer pulso de clock. También hay un tiempo mínimo para la permanencia de un dato en la entrada (la representación usada para un dato significa que el mismo puede ser alto o bajo, depende del bit que se esté grabando). Lo que no hay es un tiempo máximo pero evidentemente cuando mayor es este tiempo más lenta será la carga del programa.

El tiempo más adecuado depende de factores tales como el largo del cable utilizado para conectar el programador a la PC. Si el cable es largo, los pulsos tienden a deformarse y atenuarse, sobre todo cuando son de corta duración (100nS equivalen a una frecuencia de 10MHz). Para evitar problemas, todos los programadores trabajan a velocidades inferiores a la máxima, sobre todo considerando que la capacidad de memoria no es muy grande y para el uso no comercial no son imprescindibles grandes velocidades de grabación.

Un cargador de PIC debe permitir también una operación de verificación para cuando el usuario tiene dudas respecto de la condición de un determinado PIC (vacío o lleno). Solo que el programa almacenado debe ser inviolable en caso de que la persona que lo cargó así lo haya dispuesto. Esto se llama predisponer el PIC y no sólo se lo predispone para hacerlo inviolable, sino que además se determinan otros importantes parámetros de funcionamiento como por ejemplo que esté predispuesto para un clock a RC o que contenga un temporizador de reset interno.

Las predisposiciones no forman parte del programa del PIC aunque pueden incluirse junto con éste. Algunos cargadores de PICs preguntan sobre la predisposición antes de grabar el PIC. La pregunta sobre la predisposición deseada aparece en la pantalla de la PC y uno elige de un menú de opciones. En otros casos, las predisposiciones se escriben antes del programa en una secuencia perfectamente predeterminada que debe respetarse a ultranza. Las predisposiciones son tres y las vamos a analizar por orden:

a) *La prohibición de lectura. El PIC se programa pero el programa no puede ser leído. Cada tanto aparece información en Internet sobre alguna empresa dedicada a leer PICs protegidos, sin embargo, aún no he conseguido leer ningún PIC protegido. Aclaremos que el programa protegido no se puede leer, pero el PIC se puede volver a grabar. El autor escuchó también una especie que dice que el PIC16Fxx fue creado por una supuesta filtración en el secreto del programa de los PIC16Cxx, pero el autor no tiene pruebas de la certeza de esta noticia.*

b) *Habilitación del timer de reset. En el pinup del PIC se puede observar que la pata llamada MCLR negado (4) tiene un doble uso. En principio sirve para predisponer el PIC en grabación o lectura, como*



ya fuera comentado; pero cuando su tensión cae por debajo de 1V, el PIC se resetea y comienza la lectura del programa por el principio. Este tipo de reset se llama reset a pulsador y se utiliza para provocar alguna acción como, por ejemplo, comenzar una secuencia de encendidos de LEDs o encender un LED por un tiempo predeterminado. Este modo de funcionamiento requiere una acción externa de reset (apretar un pulsador). La misma acción de reset se puede conseguir en forma automática cada vez que se conecta la fuente de 5V. Este modo de trabajar sólo requiere que la pata MCLR (pata 4) esté permanentemente conectada a 5V (con un puente o un resistor de 1k $\Omega$ ) y que el PIC esté predispuerto con el Power-up Timer en ON (temporizador de encendido conectado). Cuando el temporizador está habilitado, luego de conectar la fuente de 5V, comienza un conteo interno (que dura 72ms) durante el cual el PIC pone todas sus compuertas internas en cero (reset) y recién después comienza a efectuar los pasos que le indica el programa. A propósito, el nombre MCLR proviene de MASTER CLEAR (literalmente, limpiador maestro).

c) Circuito de vigilancia (perro guardián). En realidad el PIC tiene un reset extra que opera luego de un cierto tiempo si no se realiza la operación final del programa. Es como una especie de rutina automática de descongelamiento que opera en caso de falla del programa. Eventualmente puede ingresar algún pulso de energía al sistema que envíe al programa a un loop (rizo) eterno (una derivación o camino cerrado sobre sí mismo). Una vez que el programa ingresó en ese camino no puede salir de él salvo que se produzca un reset. Si pasa un tiempo considerable sin que se llegue a la última sentencia del programa, el PIC analiza su predisposición y si el Watchdog timer (literalmente perro guardián) está habilitado provoca un reset programado. En los programas más simples y donde se producen loops infinitos a propósito (de los que sólo se sale tocando un pulsador), se suele dejar el perro guardián desconectado para evitar un funcionamiento errático. Cada cargador de programas tiene un software diferente y esto implica que las predisposiciones anteriores, también llamadas fusibles de predisposición se accionen con sentencias diferentes que serán explicadas más adelante.

### EL CARGADOR NOPPP

El NOPPP es un software para un cargador muy simple y efectivo. Realmente no tiene defectos importantes y nos permite cargar los PIC 16C83, 16C84 y 16F84 en forma muy económica ya que el hardware correspondiente sólo requiere dos diodos rápidos de señal, un BC548 y 4 resistores, además de una fuente regulada doble de 13 y 5V y un conector para el puerto paralelo de la PC. Más adelante veremos el circuito completo, pero para entender el funcionamiento del cargador de programas nada mejor que el circuito simplificado de la figura 4.

A propósito, el nombre del software proviene de las iniciales de NO Piece Programmer Pic, es decir: Programador para PIC sin Piezas, en alusión a los muy pocos componentes que requiere. Las patas de comunicación con el puerto paralelo de la PC están indicadas como 1J1, 2J1, etc, ya con J1 designamos al conector de entrada y el número inicial indica a qué pata se dirige el cable (también incluimos el color del mismo). Los datos (oscilando de 0 o 5V) ingresan en sucesión por el cable 14J1 y, atravesando el resistor R2, llegan a la pata 13 del PIC. La pata 13 perteneciente al puerto "B" se comporta como una pata de entrada de datos en tanto el cable 2J1 se encuentra en el estado bajo, es decir que la pata 4 Vpp está a un potencial de 13V aproximadamente.

Al mismo tiempo, el cable 17J1 se manda a potencial de masa para que los datos entrantes no sal-





gan a su vez por el cable 11J1; de este modo, el diodo D1 no permite que el potencial del cable supere la tensión de barrera del diodo (es decir que el diodo D1 opera como una llave). Como ya sabemos los datos deben ser validados por medio de un cambio de estado de la señal de clock que ingresa desde la PC por el cable 1J1.

Fórmese una imagen mental del flujo de datos. Piense en la PC como si bombeara datos al PIC por el cable 14J1; el PIC, para no inflarse, los devuelve por el cable 11J1. En realidad, los datos ingresan por la pata 13 del PIC y se instalan en la memoria; pero a continuación, el programa de carga verifica que el dato esté en la posición de memoria correspondiente y si así ocurre, se habilita la carga del siguiente dato. La secuencia es tal que:

- A) se direcciona una posición de la memoria,
- B) se graba,
- C) se verifica esta última grabación y si es correcta,
- D) se habilita al programa para cargar el siguiente dato.

De acuerdo al programa de carga, si falla la carga de un dato se puede seguir con los otros y al final se intenta la carga del dato que no se cargó. En otros, una falla de verificación significa que el programa debe volver a cargarse completo.

Ahora que conocemos el funcionamiento del cargador de programas NOPPP vamos a completar el circuito con una fuente de alimentación y el conector para PC (figura 5).

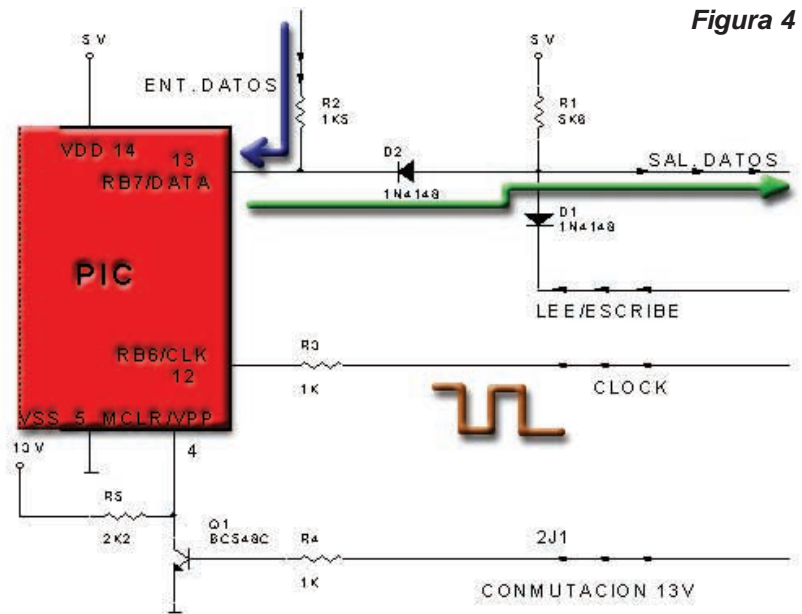


Figura 4

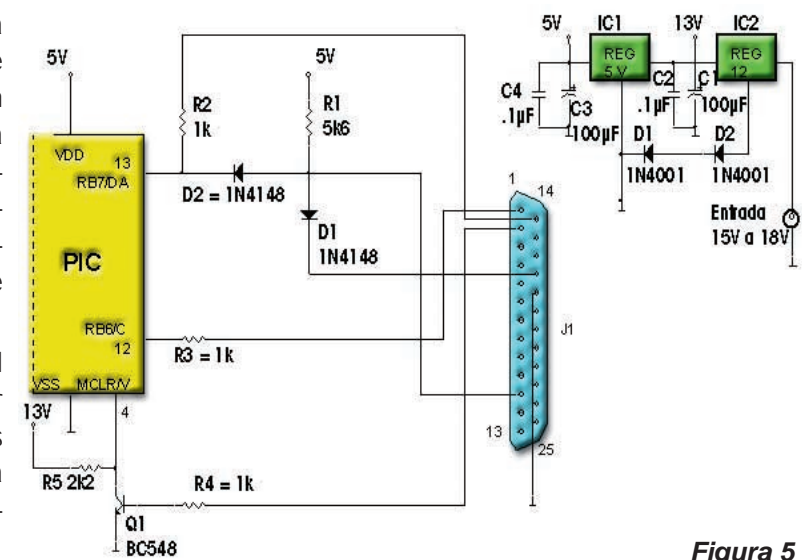


Figura 5



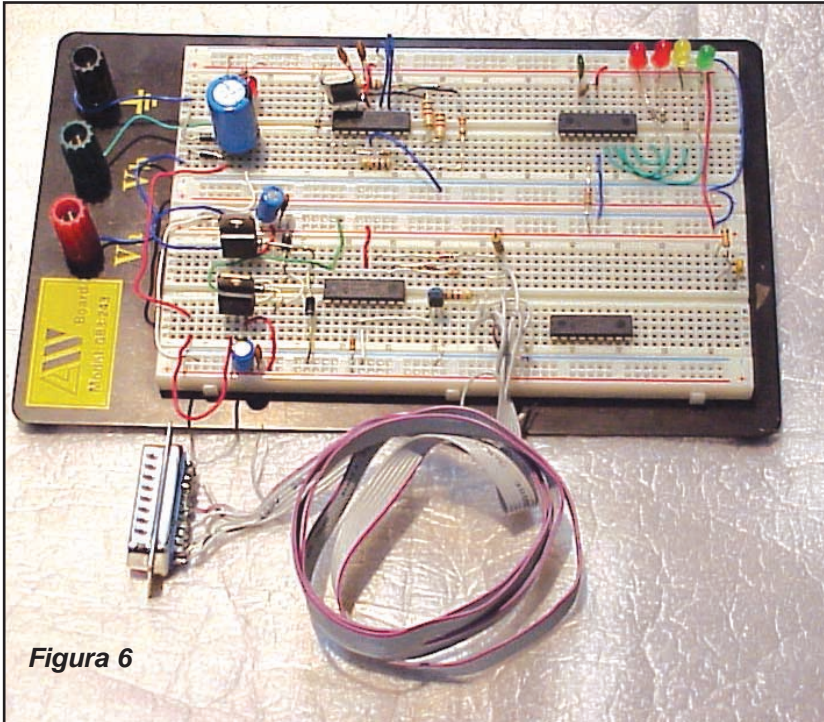


Figura 6

Observe que las fuentes están conformadas por dos reguladores de 5 y de 12V pero, como necesitamos un regulador de 13V, realizamos una pequeña modificación en el regulador de 12V; agregando los diodos D1 y D2 la tensión de salida se incrementa a un valor de 13,2V. El otro cambio importante es el agregado de capacitores de filtro que, como Ud. observa, siempre se ubican de a dos: un electrolítico para filtrar las bajas frecuencias y un cerámico disco para las altas frecuencias.

Por último, se agrega un conector del tipo DB21 macho para conectar el dispositivo directamente a la salida de la impresora de una PC. En el circuito dibujamos el conector visto por el lado de las patas de conexiones y el código de colores de cable adecuado para usar un cable plano de 5 hilos (negro, marrón, rojo, naranja y amarillo) al cual se le retuerce por encima otro de color blanco que opera como masa y blindaje. Todo este circuito es sumamente sencillo y si Ud. sólo quiere conocer los PICs mediante este manual técnico, pero no se va dedicar a trabajar permanentemente con ellos, puede armarlo en un módulo de armado sin soldaduras. En la figura 5 se observa el circuito eléctrico del cargador NOPPP, en la figura 6 se ve el armado en un protoboard, junto con el entrenador de la figura 1 y la fuente de alimentación y en la figura 6 tenemos una sugerencia para el armado en la placa de circuito impreso.

Para poder cargar un PIC con el circuito de la figura 5 empleamos el software NOPPP. Puede bajarlo desde nuestra página web [www.webelectronica.com.ar](http://www.webelectronica.com.ar), haciendo click en el ícono Password e ingresando la clave tepec26. Este software corre bajo ambiente DOS por lo cual, si tiene instalado un sistema tipo NT (como Windows XP) deberá arrancar su computadora con un disco de inicio de Windows 98.

Una vez guardado el programa en el disco rígido de su PC debe des-

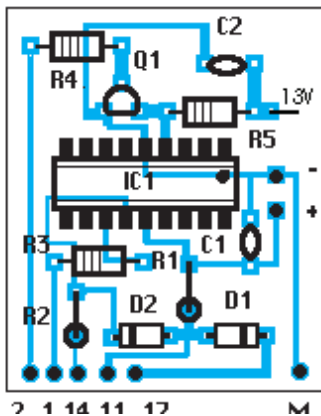
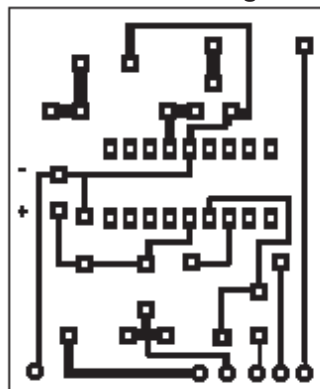


Figura 7



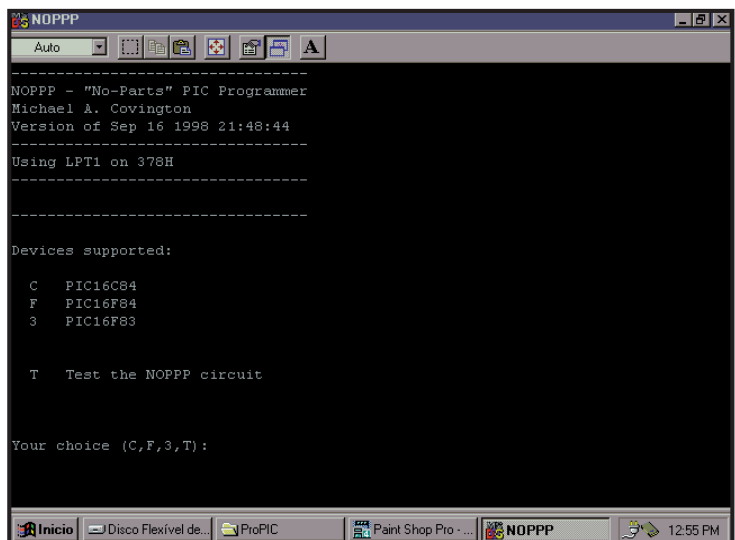
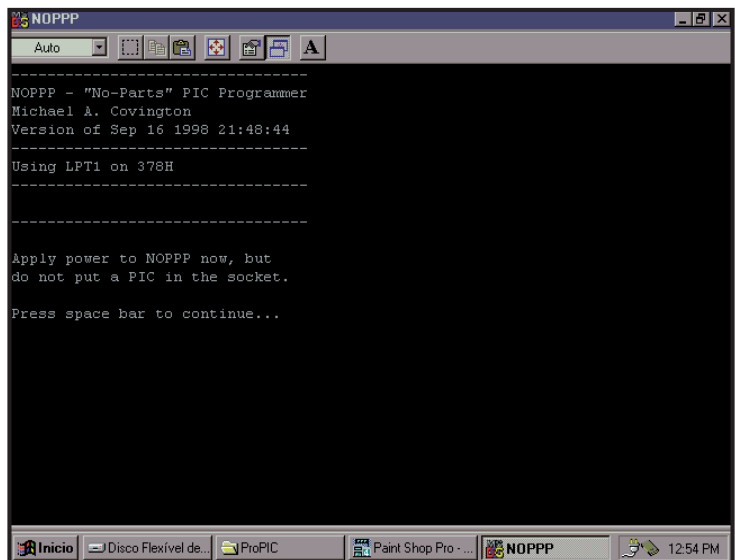
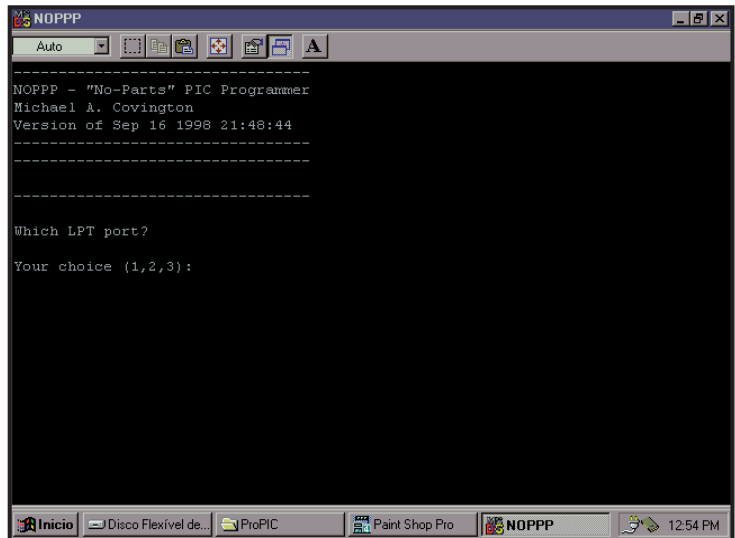
comprimirlo. Antes de ejecutarlo conecte el cargador de PICs (circuito de la figura 5 alimentado con una fuente de 5V y 13,2V respectivamente pero sin la alimentación activa) al puerto paralelo de la PC. Ejecute el archivo NOPPP del directorio PIC y aparecerá una pantalla como la que mostramos en la figura 8.

Esta pantalla inicial nos pregunta en qué puerto está conectado el hardware (circuito cargador de la figura 5) y nos da las opciones 1, 2 y 3. En la mayoría de las máquinas (salvo que se halla predispuesto lo contrario o que existan más de una salida paralelo) siempre se usa el puerto 1. Pulse la tecla "1" para pasar a la siguiente pantalla que podemos observar en la figura 9.

En esta pantalla se confirma que se está utilizando el puerto 1 y que este puerto tiene una posición de memoria 378h (para Windows 98, para otro sistema puede ser otra posición y quienes saben de computadoras también conocen la forma de modificar esta dirección si hiciera falta). Además el programa indica que se debe conectar la fuente pero manteniendo el PIC sin ubicar en el zócalo (base). Luego debemos apretar la barra espaciadora para ir a la siguiente pantalla (figura 10).

En esta pantalla el programa nos indica que sólo soporta tres tipos de PICs y debemos optar por uno de los tres, pulsando las teclas C, F o 3. También se puede hacer una verificación del funcionamiento del cargador pulsando la tecla T.

Luego de pulsar la tecla correcta (por ejemplo la C) aparece una nueva pantalla como la indicada en la



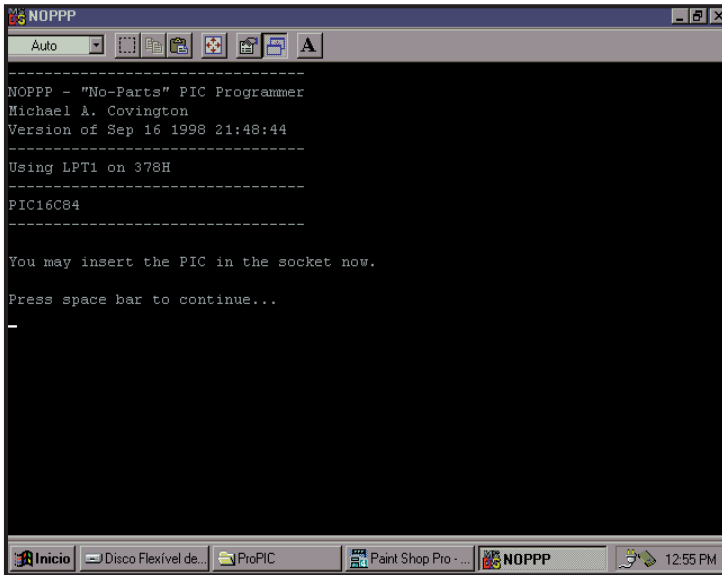
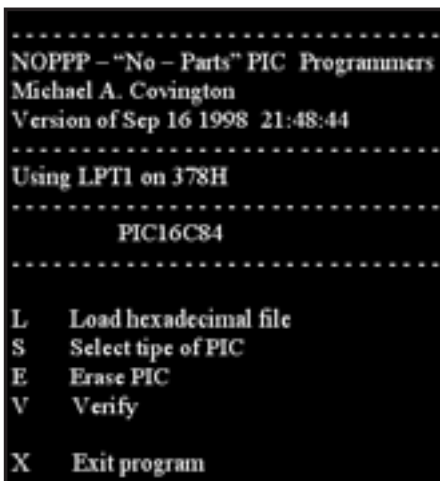


figura 11. Esta pantalla sólo nos indica que es el momento de insertar el PIC en el zócalo (note que ya aparece la palabra PIC16C84, indicando que vamos a cargar a este  $\mu$ C). Ahora debe conectar la fuente de alimentación y apretar la barra espaciadora de modo que aparezcan las opciones de manejo, tal como se muestra en la figura 12. En esta pantalla podemos elegir la operación que deseamos realizar entre las opciones (seleccionables por las teclas respectivas) L, S, E, y V que tienen las siguientes funciones:



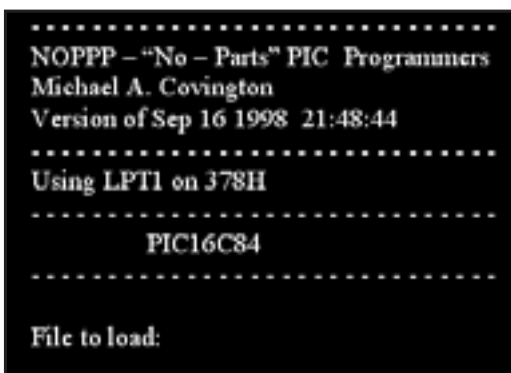
L) Cargar el PIC con un archivo que debe ser del tipo hexadecimal. Este formato no es el formato que maneja el programador por lo tanto antes de cargar el PIC debemos generar este archivo "hex".

S) Seleccionar el tipo de PIC. En nuestro caso el PIC ya fue seleccionado pero podríamos necesitar un cambio si debemos programar más de un tipo.

E) Borrar un PIC previamente grabado. En lo personal, aconsejo borrar el PIC como paso previo a la grabación por razones de seguridad.

V) Verificar el programa cargado en un PIC. En esta función se carga un programa hexadecimal y el cargador verifica que el programa almacenado en el PIC sea igual al hexadecimal.

Por lo general, el archivo de programa que tenemos disponible tiene una extensión ".asm" (más adelante veremos este tema más detalladamente) que corresponde a la versión "assembler" (ensamblado), nemotécnica o nemónica del microcontrolador con el que vamos a trabajar. Como el cargador de PIC no entiende ese idioma el programa se "pasa" por un programa traductor llamado MPASM (es un compilador) que además de la traducción realiza un control y permite depurar errores y puede bajar sin cargo de la web.



Para programar el PIC (opción P) antes debemos cargar el programa ".hex" en el NOPPP y para ello presionamos la letra "L" de "load hexadecimal file" cuya traducción exacta es "cargar archivo hexadecimal". Aparecerá una pantalla como la mostrada en la figura 13.

Escriba el nombre del archivo a cargar es decir, por ejemplo secua.hex, teniendo en cuenta que este archi-





vo debe estar dentro de la misma carpeta donde se aloja el NOPPP (si no sabe manejar DOS haga al pie de la letra lo que le indicamos). El archivo "secula.hex", que también puede bajar de nuestra web con la clave tep-ec26, corresponde a un secuenciador de 4 canales que podrá probar con el entrenador de la figura 1 de este capítulo.

Una vez tipeado el nombre del archivo apriete la tecla ENTER para que se cargue el programa en el NOPPP y el resultado de la carga se observará en una pantalla como la mostrada en la figura 14.

En esta pantalla se pueden observar algunos detalles del tamaño del archivo ".hex" desglosado en cantidad de palabras de programa, de configuración, de identificación y de datos. También aparece una indicación de que el programa fue cargado completo. Por último se indica "apretar la barra espaciadora para continuar" con lo cual aparece nuevamente la pantalla de opciones.

Ahora podemos programar el PIC eligiendo "P" (PROGRAM PIC). Se observará un movimiento en la pantalla y unos segundos después aparecerá un cartel indicando que la carga fue exitosa. Recuerde que el NOPPP cargará los datos en la memoria y luego los irá leyendo; si el último dato se cargó correctamente pasará a cargar el siguiente y así sucesivamente hasta llegar al último.

Posteriormente el programa indicará que se debe quitar la tensión de alimentación y luego sacar el PIC del zócalo. Cumpla con las indicaciones y coloque el PIC programado en el zócalo (base) del entrenador de la figura 1. Y por último alimente el circuito y observe los leds. Si todo salió bien observará que se encienden en secuencia una y otra vez.

### Prueba Preliminar del NOPPP

Si ejecuta el NOPPP y no está conectado el hardware en el puerto paralelo de la computadora, luego de apretar la tecla "1" (cuando lo solicita el programa) va a aparecer una pantalla como la indicada en la figura 15. Esta pantalla le indica que el dispositivo está desconectado (NOPPP hardware not found), si está conectado significa que su PC tiene un puerto paralelo de ida solamente (envía señales a la impresora, pero no recibe todas las señales de retorno desde la misma). En este caso ignore la pantalla que le va a aparecer siempre. Puede también aparecer esta pantalla si está mal armado el cargador o el cable

```
-----
NOPPP - "No - Parts" PIC Programmers
Michael A. Covington
Version of Sep 16 1998 21:48:44
-----
Using LPT1 on 378H
-----
PIC16C84
-----
File to load: secula.hex
Program memory loaded 17 word(s)
Configuration loaded 1 word(s)
ID memory loaded 4 word(s)
Data memory loaded 29 word(s)
Loading complete

Press space bar to continue
```

```
-----
NOPPP - "No-Parts" PIC Programmer
Michael A. Covington
Version of Sep 16 1998 21:48:44
-----
Using LPT1 on 378H
-----

Caution: NOPPP hardware not found!

with some versions of the circuit and some
parallel ports, this may be normal.

If you are sure you have chosen the correct
parallel port, press space bar to proceed.
To cancel program, press Ctrl-C.

Press space bar to continue...
```





```
-----
NOPPP - "No-Parts" PIC Programmer
Michael A. Covington
Version of Sep 16 1998 21:48:44
-----
Using LPT1 on 378H
-----

Ensure NOPPP is powered up now,
with no PIC in the socket.

Press space bar to continue...
```

```
-----
NOPPP - "No-Parts" PIC Programmer
Michael A. Covington
Version of Sep 16 1998 21:48:44
-----
Using LPT1 on 378H
-----

TEST 1

Connect negative voltmeter lead to pin 5
of PIC socket and check the following voltages:

Socket pin 4      < 0.8 V
Socket pin 12    < 0.8 V
Socket pin 13    < 0.8 V
Socket pin 14    4.75 to 5.25 V
Junction of
D1, D2, and R1  < 0.8 V

Press space bar to continue...
```

de conexión. Ignore la indicación, ya que en el proceso posterior aparecerá el modo de reparar la falla. Si desea salir del programa debe pulsar las teclas Ctrl y C al mismo tiempo o debe hacer click sobre la crucesita que está en el margen superior derecho. Para continuar, ignorando el mensaje, pulse la barra espaciadora.

La siguiente pantalla es la que ya vimos en la figura 10 donde debe elegir un determinado tipo de PIC o puede realizar el TEST de prueba del PIC. La primera vez que conecte el hardware acepte la propuesta de realizar el auto diagnóstico pulsando "T". Si recibe una pantalla como la mostrada en la figura 16 significa que el cargador no recibe tensión de fuente. Si su fuente está activa aparecerá la primer pantalla de auto diagnóstico que se observa en la figura 17.

La pantalla indica que debe conectar el terminal negativo del multímetro a GND (pata 5 del PIC) y con el terminal positivo debe medir las patas 4, 12 y 13 que deben estar a un potencial bajo menor a 0.8V y la pata 14 de fuente que debe estar a un potencial comprendido entre 4,75V y 5,25 V. La unión de los diodos D1 y

D2 con el resistor R1 debe tener una tensión menor a 0.8 V.

Estas mediciones deben realizarse sin colocar el PIC en el zócalo.

En esta prueba el cable de 13V está a potencial alto (la computadora habilita la tensión de 13V) y por lo tanto Q1 estará saturado. En estas condiciones el colector sólo puede tener menos de un volt respecto de GND y el PIC estará en condición de lectura de memoria. El cable de salida de datos tiene un potencial correspondiente a la barrera de D1 (0,6V).

Si la tensión de la pata 13 es mayor a 800mV es porque seguramente el cable conectado al terminal 17 del conector no está a potencial de masa y D1 no está conduciendo polarizado por R1. Probablemente esté cortado el cable, o está mal conectado a la pata 17 del conector J1. También puede ocurrir que el cable esté a potencial de masa pero el diodo D1 esté abierto o invertido. Para sacar conclusiones más adecuadas verifique la tensión de la unión de los diodos D1 y D2.

```
-----
NOPPP - "No-Parts" PIC Programmer
Michael A. Covington
Version of Sep 16 1998 21:48:44
-----
Using LPT1 on 378H
-----

TEST 2

With negative voltmeter lead still
connected to pin 5 in the PIC socket,
check the following voltages:

Socket pin 4      12.0 - 14.0 V
Socket pin 12    > 4.0 V
Socket pin 13    > 4.0 V
Junction of
D1, D2, and R1  < 0.8 V

Press space bar to continue...
```

Si Ud. tiene una tensión en la pata 4 (tensión de grabación) alta (cercana a los 13V) probablemente tendrá cortado el cable que va conectado a la pata 2 del conector J1. También puede ocurrir que el transistor Q1 esté abierto o no sea de la polaridad adecuada (esto también se presenta si por error conecta al revés el transistor).

Si tiene tensiones altas en las patas 12 y 13 deberá



buscar cortocircuitos en las patas del zócalo (base) del PIC o en el circuito impreso o en el conector J1. Si las tensiones medidas son correctas, presione la barra espaciadora para pasar al test 2. Aparecerá una pantalla como la indicada en la figura 18.

Debe medir las tensiones de las patas 4, 12 y 13 con respecto a tierra o masa (pata 5) pero esta vez con el transistor Q1 cortado.

Las tensiones medidas en las patas 12 y 13 provienen de la computadora y dependen de su fuente regulada interna y del puerto de salida paralelo. Lo importante en esta prueba es que la tensión supere los 4V para que el PIC interprete que hay un estado alto en su entrada de DATOS y en el CLOCK.

En el test 1 se verificó el estado bajo de estas patas y en éste se verifica el estado alto completando la prueba de las patas 12 y 13. La tensión medida en la pata 4 debe tener un valor comprendido entre 12 y 14 V aproximadamente, ya que el transistor Q1 no debe conducir (dependiendo de la tensión que entregue nuestra fuente de alimentación, ya no la computadora).

La unión de los diodos D1 y D2 deberá mantenerse a un potencial inferior a los 800mV asegurándonos que el diodo D2 no está en cortocircuito o invertido. Observe que en este test el diodo queda polarizado en inversa.

```
-----
NOPPP - "No-Parts" PIC Programmer
Michael A. Covington
Version of Sep 16 1998 21:48:44
-----
Using LPT1 on 378H
-----

TEST 3

with negative voltmeter lead still
connected to pin 5 in the PIC socket,
verify that pin 13 > 4.0 V.

Next, insert a 470-ohm resistor into the socket
connecting pin 13 to pin 5 and verify that
pin 13 drops to < 0.7 V with the resistor in place.

Press space bar to continue...
```

Si la tensión de la pata 4 es baja o no existe, el resistor R5 está cortado, mal conectado o con un valor equivocado; también puede ocurrir que el transistor tenga un cortocircuito entre el emisor y el colector (puede desconectarlo del circuito para verificar que suba la tensión de fuente). Debe verificar primero que la fuente de 13V tenga la tensión correcta. Si la pata 12 no tienen una tensión superior a 4V, probablemente el resistor R3 esté equivocado de valor o mal conectado. Si la tensión en la pata 13 es menor a 4V puede estar equivocado o abierto el resistor R2

o el diodo D2 puede estar invertido o en cortocircuito.

Si las tensiones son correctas presione la barra espaciadora para pasar al test 3 (aparecerá la pantalla de la figura 19) en donde se simulará un estado bajo saliendo de la pata de dato del PIC. La acción del PIC para sacar datos se simula conectado un

```
-----
NOPPP - "No-Parts" PIC Programmer
Michael A. Covington
Version of Sep 16 1998 21:48:44
-----
Using LPT1 on 378H
-----

TEST 4

with negative voltmeter lead still
connected to pin 5 in the PIC socket,
verify that the junction of D1, D2, and R1
is < 0.8 V.

Press space bar to continue...
```

resistor de 470 $\Omega$  entre las patas 13 (RB7/DATA) y la pata 5 (masa). La tensión de la pata 13 debe ser ahora, inferior a 0.7V al apoyar el resistor.

Si las mediciones son correctas podemos pasar al test 4. Apretando la barra espaciadora aparece una pantalla como la indicada en la figura 20.

Es este test se establece el estado bajo de la salida de datos cuando se escribe un dato alto. Esto significa que no debe haber retorno directo por el cable de escritura debido a que el diodo D1 está polarizado con



su cátodo a masa para evitar que la tensión sobre él supere los 600mV. Si la tensión es la correcta, pulse la barra espaciadora para pasar a la última pantalla de test . Aquí se prueba al dispositivo como generador de un dato alto. Si todas las pruebas pasaron correctamente se puede asegurar que ya no hay errores de armado y que los diodos y resistores no están alterados.

### EL CARGADOR QUARK PRO 2

En la revista Saber Electrónica y en este texto, se presentan muchos cargadores, sin embargo, desarrollamos este prototipo para usuarios de PICs más exigentes. Si bien los cargadores publicados han sido de mucha utilidad, este nuevo prototipo les permitirá programar PICs más poderosos como el PIC16F873 (de 28 pins) , el PIC16F874 (de 40 pins) o pics mejorados como el PIC16F627.

Existen en Internet muchos programadores de pics de uso libre, como el JDM, el TAIT, el PROPIC 2 y el TAFE, entre otros muchos. Sin embargo, ningún quemador (programador) funciona sin un programa que lo opere, para ello existen también muchos utilitarios como el IC-PROG, el PIC-ALL, el NOPPP, el PONY-PROG, etc. (muchos de uso libre). Cada programador y cada programa posee sus propias ventajas y desventajas, lo importante es escoger un programa que sea compatible con el programador elegido.

#### *¿Por qué basar el diseño en el programador JDM?*

En principio, para programar un PIC se requiere de una fuente de 5 volt para alimentarlo (Vdd) y otra fuente de 13 volt para programar o "quemar" el PIC (Vpp) lo que implica que debemos tener dos fuentes de voltaje para realizar la operación, lo que de entrada nos elevaría el costo.

En el JDM (véase [www.jdm.homepage.dk](http://www.jdm.homepage.dk)) cabe destacar "lo ingenioso del diseño" ya que emplea los voltajes disponibles del puerto serie de la computadora para que, mediante un arreglo con diodos y capacitores (llámese multiplicador de tensión integrado por D3, D4, D5, D6, C1 Y C2) se puedan obtener los más de 13V necesarios para programar cualquier PIC. Esto nos ofrece una gran ventaja, ya que no requerimos fuente de alimentación externa alguna convirtiendo el programador en un circuito "parásito" que en nuestro caso se alimenta del puerto serie de la computadora, logrando así abatir el costo más alto que representan las fuentes de alimentación.

Otra ventaja que ofrece este programador es que no requerimos de un limitador de corriente para el caso de insertar un PIC dañado, ya que el circuito RS232 del puerto serie de la PC tiene su propio limitador de corriente. Sin embargo, no podemos tener toda la gloria, ya que para los que se dedican a la programación "in situ", es decir, los que programan los PIC's en el mismo circuito donde opera y el cual se diseña con un conector especialmente destinado para la programación, este programador o cargador de PIC's no funciona ya que la tierra del circuito no es compatible con la tierra flotante de nuestro programador. Para tal caso, se supone que los que realizan este tipo de programación se dedican profesionalmente a ello, y por ende contarán también con un programador profesional que tenga sus propias fuentes de alimentación. Cabe recordar que el objetivo principal de nuestro prototipo es que sea económico, fácil de usar y confiable.

En la figura 21 se muestra el diagrama esquemático del programador JDM. En la figura 22 se muestra en circuito impreso sugerido.



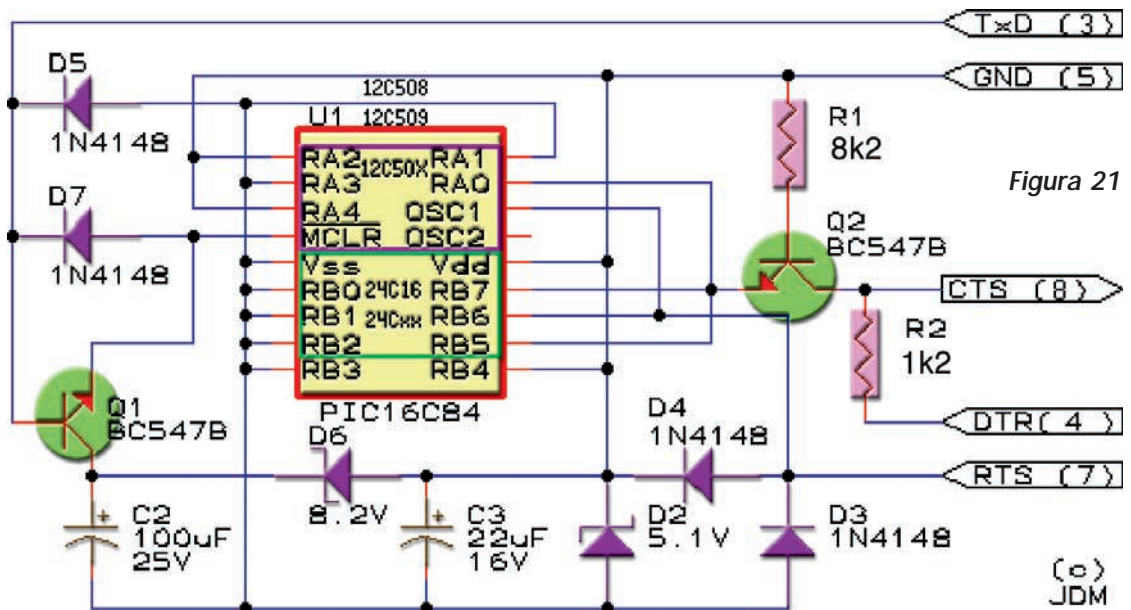


Figura 21

(c) JDM

*¿Por qué utilizar el programa IC-PROG?*

Con el mismo criterio que seleccionamos el hardware (programador JDM), seleccionamos el software (IC-PROG), basados en el que fuera más compatible. El IC-PROG ofrece varias ventajas: La primera es que es dentro de su menú ofrece opciones importantes como la posibilidad de seleccionar el puerto a utilizar, así como el prototipo de programador a utilizar; la segunda es que el programa ofrece un ambiente de trabajo muy amigable, ya que este programa y gracias a las aportaciones de muchos colaboradores de todo el mundo, está traducido a varios idiomas y tercero es que es compatible con la mayoría de los sistemas operativos de la PC, además que con frecuencia están disponibles de manera gratuita versiones actualizadas (véase [www.ic-prog.com](http://www.ic-prog.com)). En la figura 23 se muestra el ambiente de trabajo de este programa (aclaramos que en el próximo capítulo explicaremos mejor el uso de este programa y las funciones que presenta). Si usted ha trabajado con ambientes de programas diferentes, podrá observar que el ambiente del IC-Prog dispone de herramientas de trabajo muy completas.

Para cargar un programa en un microcontrolador PIC se requiere de una serie de

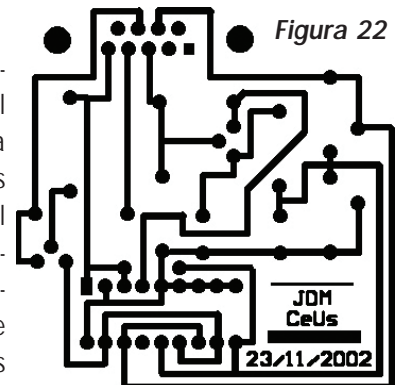
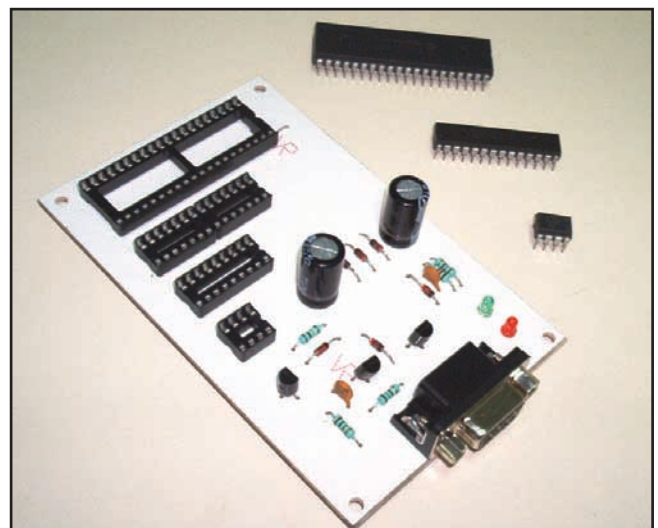


Figura 22





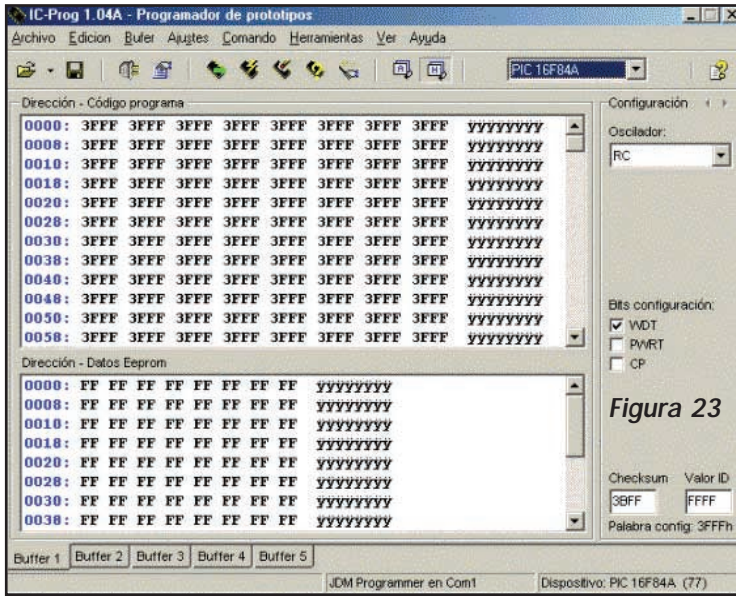


Figura 23

parámetros básicos, tal como hemos visto y el Quark Pro 2 no es la excepción. Las tensiones necesarias son:

VDD = Voltaje de alimentación de 5V

VSS = Referencia de tierra del circuito

VPP = Voltaje de programación de 14V

PGD = Datos de programación

PGC = Pulsos de reloj para la sincronización

En la figura 24 se muestra el circuito eléctrico del programador QUARK-PRO 2.

Como puede observar no existe ninguna fuente de alimentación externa. El circuito se alimenta del puerto serial de la PC a través del conector DB9. El voltaje de alimentación VDD se obtiene de los mismos pulsos de reloj (pin 7 del DB9), los cuales son rectificadas por los diodos D3 y D4 y estabilizado a 5 volt mediante el diodo zener D5 (5.1V) y el capacitor C1. De la misma manera, se obtiene el voltaje de programación VPP, cargando el capacitor C2 y estabilizando con el diodo zener D6 (8.2V) el cual se suma al voltaje del zener D5 (5,1V), obteniéndose así 13.3 Volt suficientes para reali-

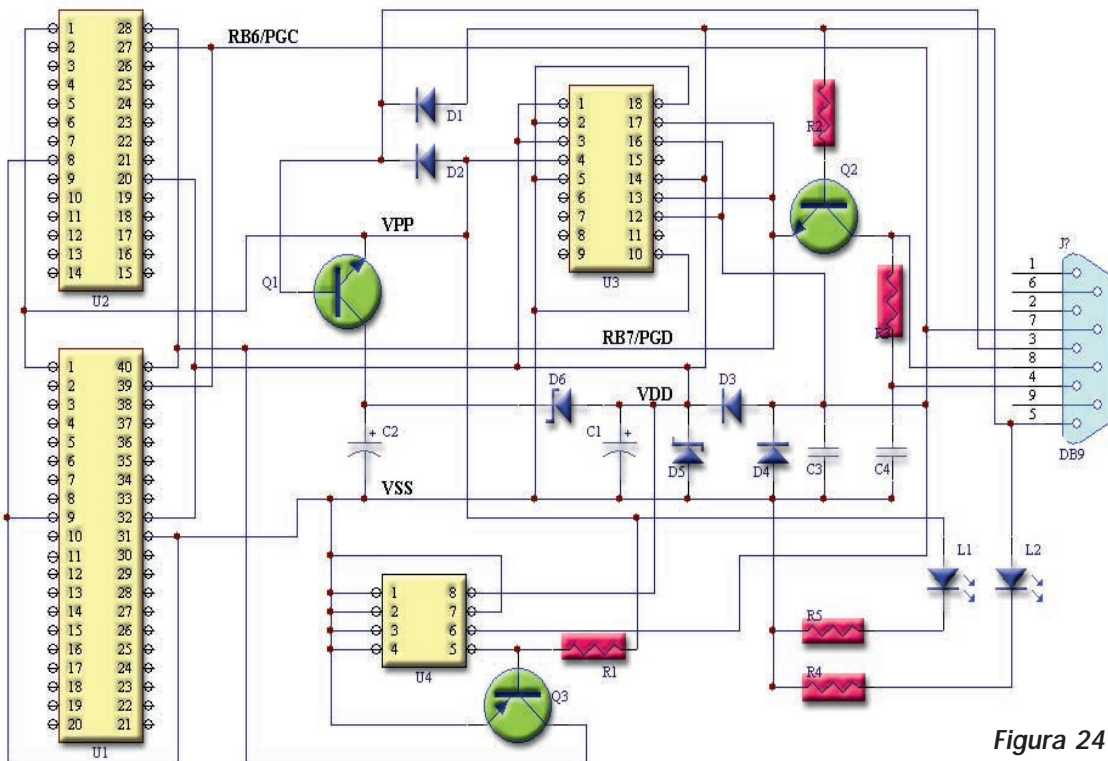


Figura 24





zar la programación del PIC. Se han adicionado el LED L1 para visualizar el proceso de grabación o lectura del PIC, así como el LED L2 para indicar que el circuito se encuentra alimentado, además, si se colocara un PIC en corto, este LED se apagará o bajará significativamente su intensidad.

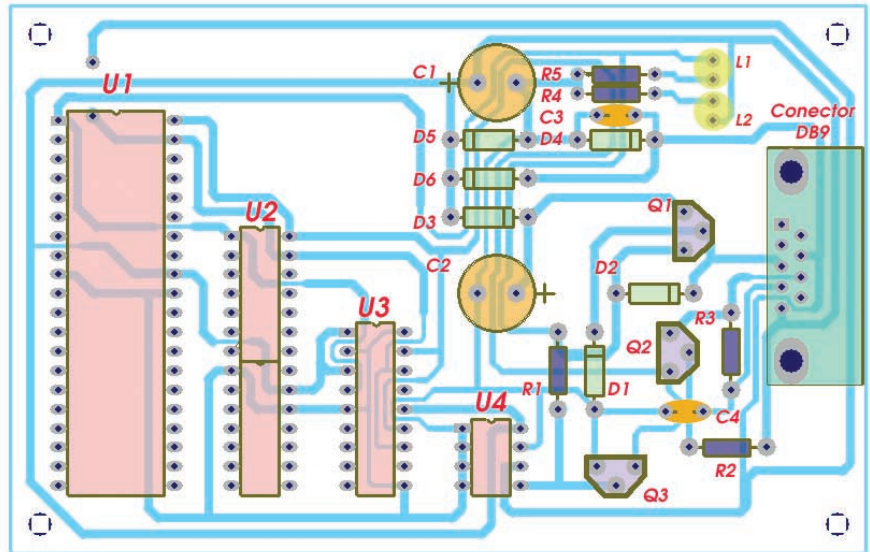
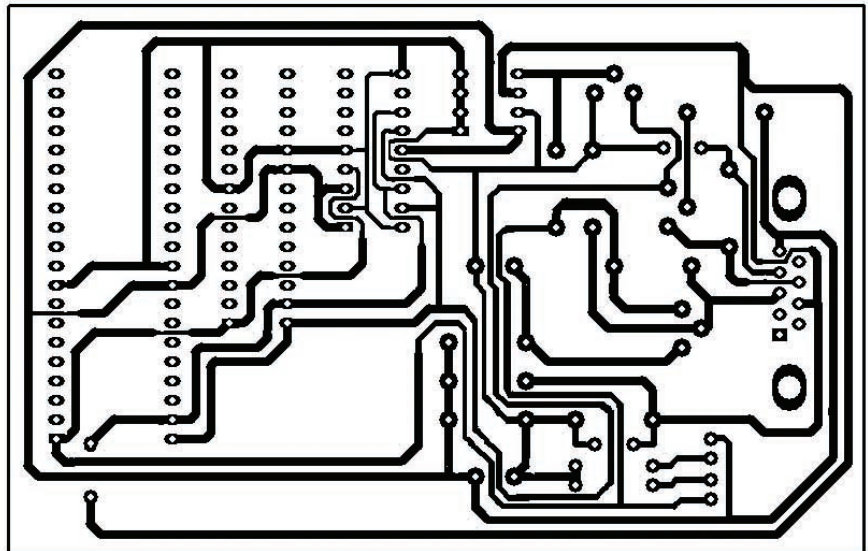


Figura 25

Más adelante explicaremos en detalle el proceso completo de transmisión (grabación) y recepción (lectura) de datos, así como la función de cada uno de los componentes del programador.



En la figura 25 se muestra el circuito impreso sugerido para nuestro prototipo.

Una vez que haya armado su prototipo de programador (quemador) QUARK-PRO, ya tiene listo su hardware, ahora le falta el software, que para nuestro caso hemos sugerido el IC-PROG.

Vaya a la página de Internet [www.ic-prog.com](http://www.ic-prog.com) e ingrese al área de descargas (download), ahí encontrará varias versiones de este programa tal como se sugiere en la figura 26.

Descargue la última versión (a la fecha de redactar este artículo era la versión 1.05C) y guárdela en una carpeta

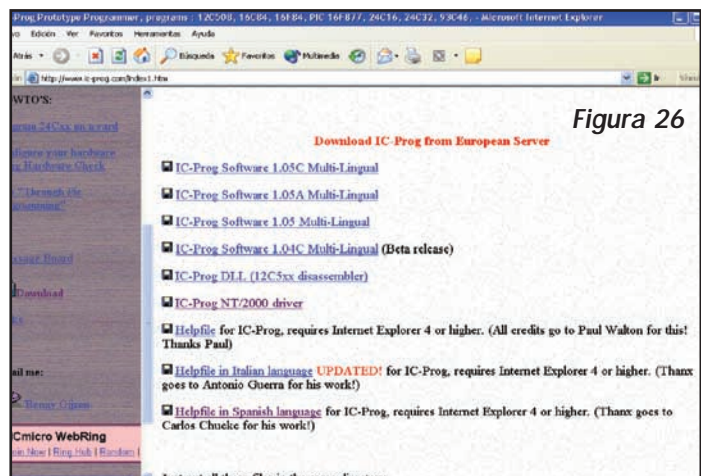


Figura 26





Figura 27



Figura 28

de su disco rígido. También descargue el archivo IC-Prog NT/2000 driver y guárdelo en la misma carpeta, porque puede serle útil en caso de que esté usando un sistema operativo Windows XP y tenga conflictos en la ejecución del IC-Prog.

Abra el archivo con Win-Zip y ejecute el programa icprog.exe (figura 27) para instalarlo. Una vez que haya hecho esto, aparecerá un ícono de acceso directo a su programa (figura 28) que le permitirá abrir directamente en programa cuando lo desee utilizar. Al ejecutar este ícono aparecerá una ventana del programa (figura 23).

Ahora solo queda probar su proyecto. Conecte al puerto serial de su PC el programador y el led rojo (L2) deberá encender. Seleccione cualquier dispositivo y elija la opción "programar", ejecute y observe el led verde (L1). Este debe

encender conforme se lleve a cabo el proceso. Si todo va bien: ¡felicidades!, su proyecto funciona. Si tiene algún problema o duda, por favor escriba a [ateclien@webelectronica.com.ar](mailto:ateclien@webelectronica.com.ar) y con gusto lo ayudaremos. Recuerde que en el próximo capítulo explicaremos mejor las funciones del Icplog, cuando detallemos cómo hacer un proyecto "desde cero".

A continuación damos la lista de materiales de este cargador:

- U1 - Base (zócalo) para CI de 8 pins
- U2 - Base para CI de 18 pins
- U3 - Base para CI de 28 pins (usar 2 bases de 14 pins)
- U4 - Base para CI de 40 pins
- DB9 - Conector DB9 hembra
- D1, D2, D3, D4 - Diodo 1N4148
- D5 - Diodo zener de 5.1 V 1W
- D6 - Diodo zener de 8.2 V 1W
- Q1, Q2 - Transistor BC547C
- Q3 - Transistor BC557C
- C1, C2 - Electrolíticos de 470 $\mu$ F x 25 V
- C3, C4 - Cerámicos de 0.001 $\mu$ F



R1 - Resistencia de 100k $\Omega$

R2 - Resistencia de 10k $\Omega$

R3 - Resistencia de 1.5k $\Omega$

R4 - Resistencia de 1k $\Omega$

R5 - Resistencia de 2.7k $\Omega$

L1 - Diodo Led verde

L2 - Diodo Led rojo

Si lo prefiere puede solicitar el montaje completo armado o el kit para que Ud. lo arme, los cuales se proveen con el instructivo de trabajo y un CD completísimo, con un Curso de Microcontroladores PIC, varios prototipos completos, un curso de programación, un montón de programas, un clip completo en VCD, además incluimos la última versión del programa de desarrollo para PIC'S MPLAB de Microchip, a entornos de Desarrollo (MPLAB, MPASM, etc.) y gran cantidad de aplicaciones. El costo del kit con el prototipo armado en Argentina es de \$95 y en México de \$290 y el kit con la placa y los componentes para montar tiene un valor en Argentina de \$85 y en México de \$260. Si está interesado en estos productos, puede solicitar mayor información al mail: [ateclien@webelectronica.com.ar](mailto:ateclien@webelectronica.com.ar) (por teléfono en Argentina llame al (011) 4301-8804 y en México llame al teléfono: (0155) 5839-7277).

### OTROS CARGADORES

En varias direcciones de internet aparece el circuito que describiremos a continuación, adjudicándole la autoría a diferentes personas y con el nombre "Cargador Universal". La versión que reproducimos es en base al trabajo de un tal "Droky" de RaDiKAI ByTEs. Nuestros técnicos han realizado algunas modificaciones y el resultado es el siguiente:

Con este proyecto se puede realizar un programador para PICs y EEPROMs de la serie 24Lxx que funciona bien y requiere pocos componentes externos. El circuito está basado en el "Luddi" o programador de JDM (vea: [www.jdm.com](http://www.jdm.com)), siendo compatible con diferentes softwares cargadores. Se ha probado en diferentes computadoras desde un Pentium 100 hasta un Pentium III de 850MHz sin inconvenientes. No requiere de ningún tipo de adaptador, y en tres zócalos que dispone, podemos programar:

- PIC12C508 /A, PIC12C509 /A
- PIC12C67x
- PIC16C55x
- PIC16C61
- PIC16C62x
- PIC16C71
- PIC16C71x
- PIC16C8x
- PIC16F8x
- PIC16F873/4/6/7 (Modelos de 28 patas)



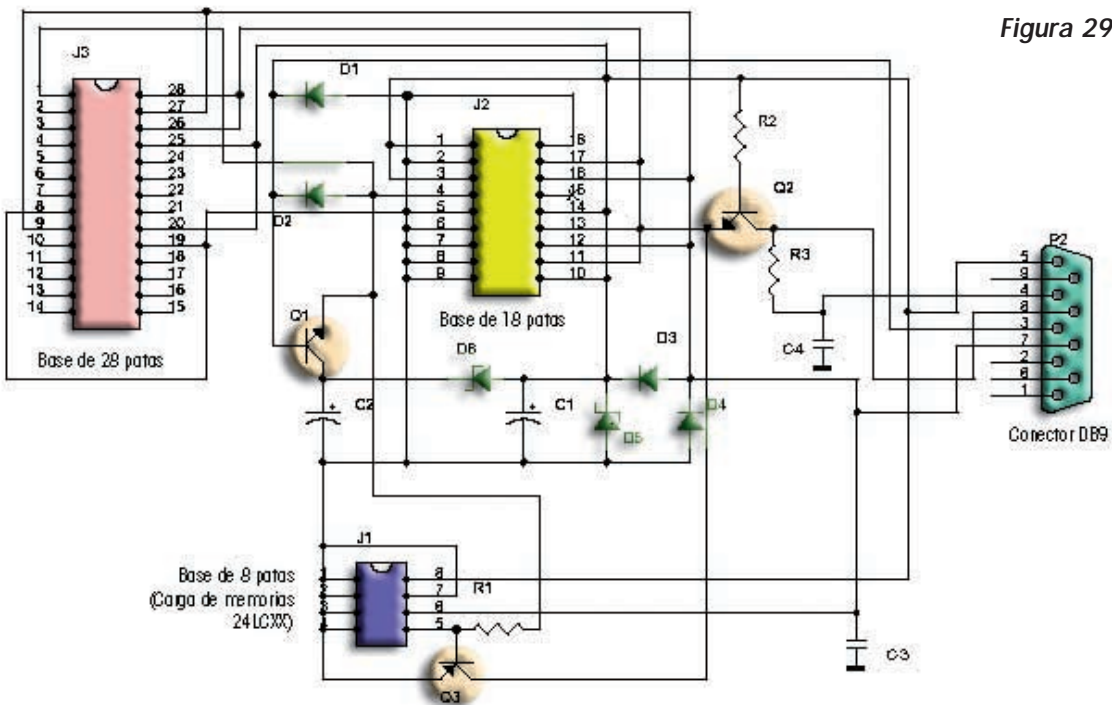


Figura 29

- PIC16C73B/74B/76/77
- EEPROMs de la serie 24LCxx

En la figura 29 se muestra el circuito del cargador (note el parecido con el Quark Pro 2 descrito anteriormente) y en la 30 la placa de circuito impreso propuesta. El tamaño del cargador es bastante reducido y posee bases (zócalos) para la mayoría de los chips que puede programar (siempre en encapsulado DIL no en TSOP, SMD ni otros formatos).

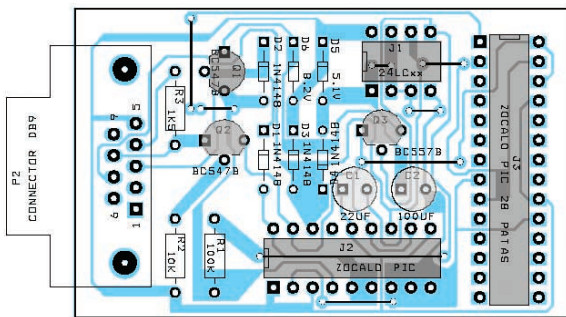
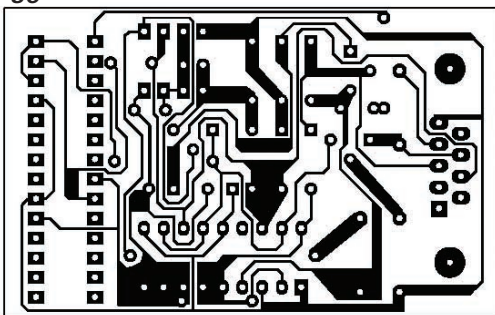


Figura 30



Si bien es posible utilizar distintos tipos de softwares, hemos realizado experiencias con el italque, el desden, el pix y el icprog. Es precisamente el icprog el que permite "cargar" a casi todos los chips indicados y lo puede bajar gratuitamente siguiendo los pasos que mencionamos anteriormente, utilizando la clave de acceso: aiwa15.

El programa pix se puede usar para los PIC16F8x y PIC16C8x, es rápido y permite bastantes opciones. Para las EEPROM 24LCxx, pueden emplearse tanto el Icprog como el PIX. Para los 12C508/A y 12C509 /A lo más se-





guro es usar el prog508 y el prog509 (vea los foros PSX) y que se pueden descargar de la página de JDM.

La lista de materiales del cargador universal es la siguiente:

- C1 - 22 $\mu$ F , 16v Tantalio/Electrolítico
- C2 - 100 $\mu$ F , 16v Tantalio/Electrolítico
- C3, C4 - 0,001 $\mu$ F - Cerámico
- D1,D2,D3,D4 - 1N4148
- D5 - 5.1V 1/2W
- D6 - 8.2V 1/2W
- J1 - Zócalo 24LCXX 8 Patillas
- J2 - Zócalo PIC 18 Patillas
- J3 - Zócalo PIC 28 Patillas
- P2 - Conector DB9 Hembra
- Q1,Q2 - BC548B
- Q3 - BC558B
- R1 - 100k $\Omega$
- R2 - 10k $\Omega$
- R3 - 1k5

Ahora bien, como hemos dicho, existen muchos circuitos cargadores de PICs que se pueden bajar de Internet.

En la figura 31 se observa el PIC800 que es un programador para 16C84, 16F84, 16F873/74/76/77 que está presentado en la web por Miguel Scapolla ([www.miguelo.com.ar](http://www.miguelo.com.ar)).

El PICPRO es un programador avanzado en relación con el NOPPP y sirve para ser montado cuando no se quieren correr riesgos con la PC que se está utilizando para cargar PICs y cuando se desea tener un trabajo libre de errores (el NOPPP es un programa para aprendizaje). Su circuito se lo puede ver en la figura 32. Por último, el PP84 es un programador sencillo del tipo NOPPP de buenas presta-

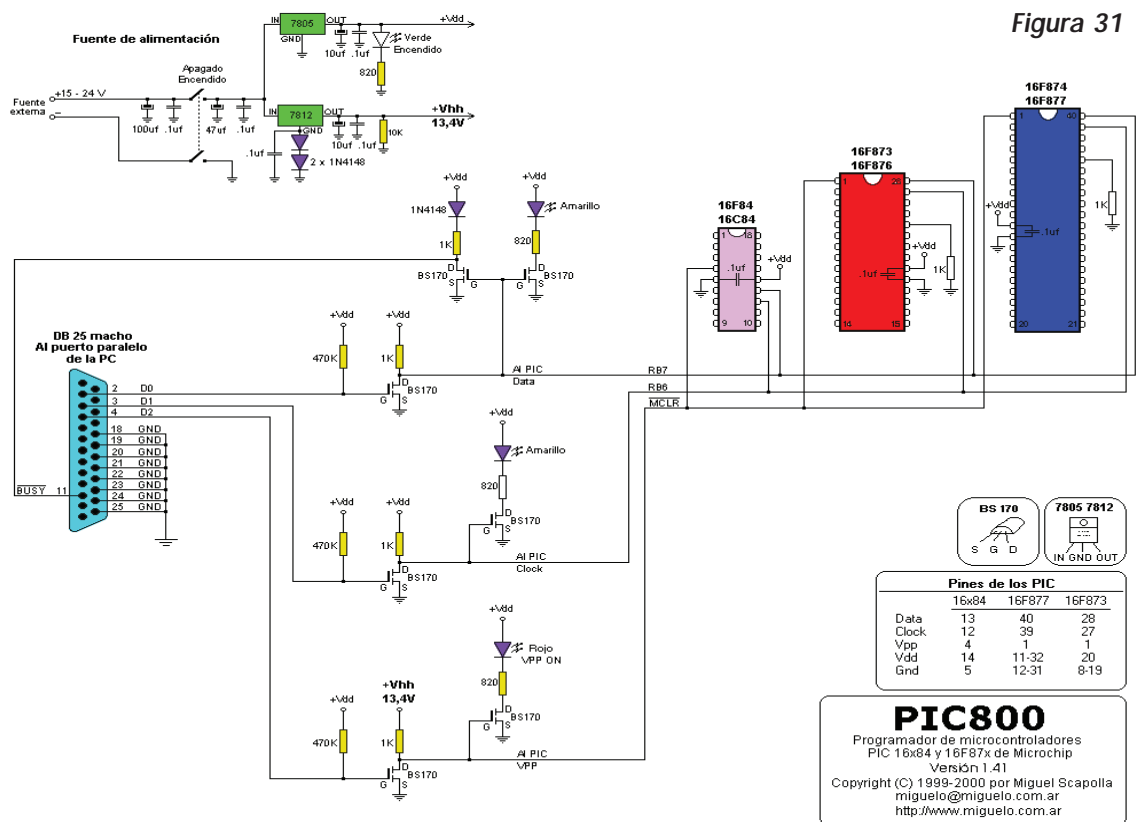
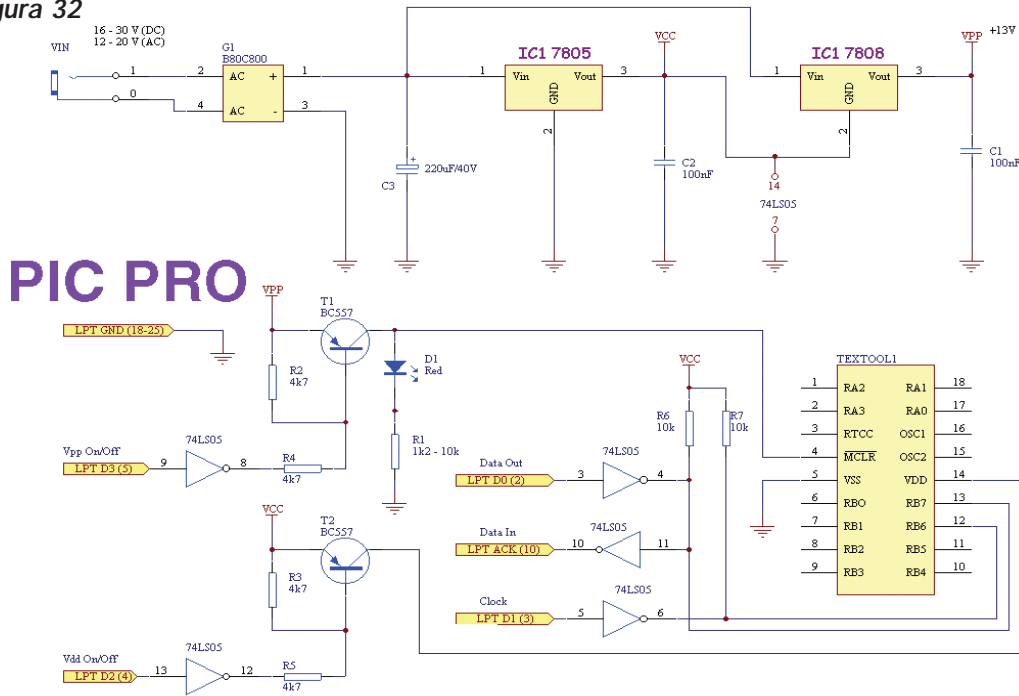


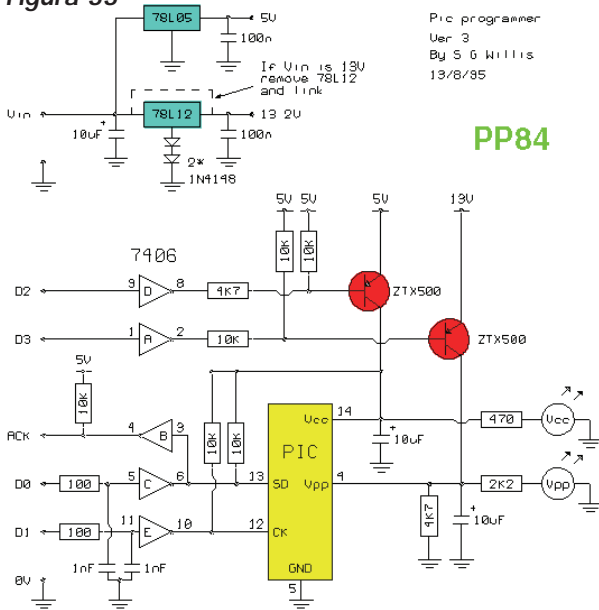


Figura 32



PIC PRO

Figura 33



Pic programmer  
Ver 3  
By S G Willis  
13/8/95

PP84

ciones (figura 33). Si desea obtener los programas que permitan manejar a estos cargadores, puede buscarlos en Internet o consultar a: [ateclien@webelectronica.com.ar](mailto:ateclien@webelectronica.com.ar)

### Conclusiones del Capítulo 3

Los microcontroladores PIC como el 16F84, precisan un potencial de tierra (pata 5), una tensión de alimentación de 5V (pata 14) y una tensión de predisposición de 13V (pata 4) para que se pueda programar su memoria EEPROM; los datos ingresan por la pata 13 en sincronismo con una señal de reloj aplicada en la pata 12. Los denominados "cargadores" efectúan estas operaciones, empleando el puerto serie o el puerto paralelo de una computadora. Existen muchos circuitos entre los que destacamos al cargador NOPPP (que emplea el puerto paralelo de una PC) y el Quark Pro 2 que se conecta al puerto serie. La ventaja del Quark Pro 2 radica en que no precisa fuente de alimentación externa (usa la tensión del puerto de la computadora) y que permite cargar a una gran cantidad de microcontroladores y memorias EEPROM.

\*\*\*\*\*

