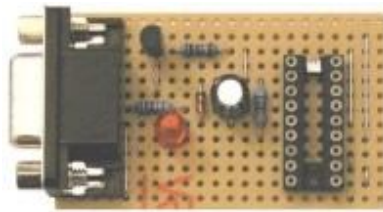


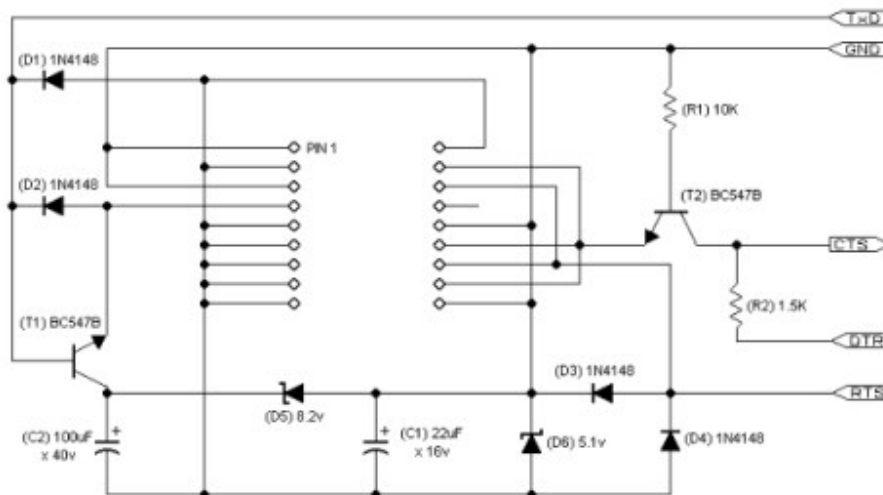
Programador para pic

Los microcontroladores son la solución a casi cualquier problema de diseño en el campo de la electrónica digital. Es indispensable tener a mano una herramienta que nos permita programarlos.



Efectivamente, los microcontroladores en general, y los de la empresa Microchip en particular, necesitan de un circuito electrónico auxiliar que nos permita transferirles desde el ordenador el programa que hemos escrito para ellos.

Hay muchas maneras de encarar este problema, y en general se suele utilizar alguno de los puertos disponibles en cualquier ordenador para este fin. Así es que en el mercado es posible conseguir “quemadores” de PICs con conexión para puerto USB, paralelo o serie (RS-232). Por razones de simplicidad en el diseño y por ser seguramente una de las alternativas más económicas, es que en este artículo vamos a explicar paso a paso como construir el programador JDM, con conexión serial. No solo nos permitirá grabar datos en los micros más comunes, de 8, 18, 28 y 40 pines, sino que también lo podremos utilizar para grabar y leer varios tipos de memorias.



Los microcontroladores de Microchip (PICs) se programan mediante un protocolo tipo serie. Se necesitan dos tensiones de alimentación para poder llevar a cabo la programación: una de 4.5v a 5.5v (VDD) y otra comprendida entre 12v y 14v (VPP), que es la que indica al PIC que va a ser programado, para que el cambie la función que realizan los pines I/O implicados en la programación. Los pines implicados en la programación varían de un microcontrolador a otro, pero en general, los de un mismo numero de pines (8, 18, etc.) tienen las mismas patitas asignadas a la programación, lo que nos permite construir programadores que sirvan para mas de un PIC. En el caso del 16F84A, 16F628A y casi todos los PIC más populares de 18 pines, se utilizan los siguientes pines durante la programación, y son los que se indican en la tabla al final de esta página.

Es muy recomendable leer la hoja de datos de Microchip sobre este tema (son unas 16 paginas, y lo encuentran en los links que recomendamos en este articulo) dado que es conveniente conocer el mecanismo implicado en la programación (tiempos, señales, etc). El documento se refiere específicamente a la familia 16F8x, pero los demás no difieren demasiado de lo allí expuesto. Con estos conceptos en mente, podemos pasar a la construcción de nuestro circuito programador, que será el encargado de transferir el programa que escribamos en la PC a la memoria FLASH del PIC. Esta es una memoria no volátil, de bajo consumo, que se puede escribir y borrar en el circuito integrado (al igual que las EEPROM). Microchip comercializa dos microcontroladores prácticamente iguales que sólo se diferencian en que la memoria de programa de uno de ellos es tipo EEPROM y la del otro tipo Flash. Se trata del PIC16C84 y el PIC16F84, respectivamente. Además de esta memoria, casi todos los PICs también disponen de una memoria de datos de lectura y escritura no volátil, (esta del tipo EEPROM). De esta forma, un corte en el suministro de la alimentación no ocasiona la pérdida de la información, que está disponible al reiniciarse el programa. Por ejemplo, el 16F84 dispone de 64 bytes de memoria EEPROM para contener datos, y los programas que creemos pueden leer y escribir en ella.

Pin	Nombre	Durante la programación		
		Función	Tipo	Descripción
12	RB6	Clock	E	Pulsos de reloj
13	RB7	Data	E/S	E/S de datos
4	MCLR	VTest Mode	P	Selección de modo
14	VDD	VDD	P	Positivo (Vcc)
5	VSS	VSS	P	Masa (GND)

Pines utilizados durante la grabación.

La construcción de un programador de PICs puede ser un proyecto muy simple o muy complicado, de acuerdo a las características que necesitemos incluir en el. En nuestro caso, dado que estamos desarrollando un circuito que sea lo mas didáctico posible, vamos a intentar mantener las cosas lo mas simples posible.

Si nos damos una vuelteita por internet, vamos a ver infinidad de circuitos programadores, algunos con alimentación externa y otros no, por puerto serial, paralelo o USB, para un solo modelo de PIC o para varios, etc.

El programador que construiremos se conoce como “JDM”, por las iniciales de su creador (Jens Dyekjær Madsen). Existen muchas variantes de el, pero básicamente todas tienen características muy similares. Es un circuito muy simple, pero que tiene varias ventajas que lo hacen muy interesante:

- >Se conecta al puerto serie, que generalmente en cualquier PC esta disponible.
- >Existe software gratis para utilizarlo, incluso bajo DOS, LINUX y por supuesto Windows (incluido WinXP)
- >Sirve para programar varios modelos de PICS (PIC12C5XX, 12C67X, 24CXX, 16C55X, 16C61, 16C62X, 16C71, 16C71X, 16C8X, 16F8X entre otros) y también para leer/escribir varios chips de memoria (24Cxx). Otros microcontroladores también pueden ser programados mediante un adaptador.
- >Dispone del conector ICSP (In-Circuit Serial Programming) para la programación de microcontroladores sin necesidad de desmontarlos de su placa de circuito impreso.
- >No necesitamos de una fuente de alimentación externa, ya que se alimenta directamente del puerto de la PC.
- >Su costo es muy bajo, los componentes necesarios difícilmente nos cuesten mas de 3 o 4 u\$s y son muy fáciles de conseguir.

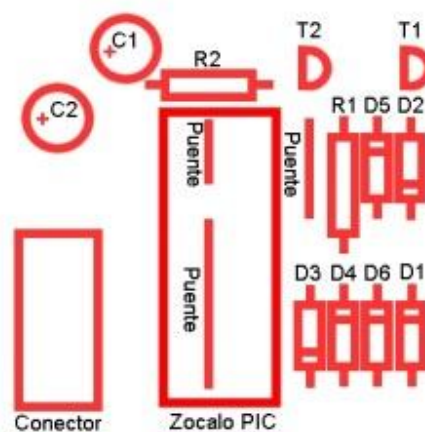
Una aclaración importante antes de seguir adelante: el hecho de que el programador se conecte a un puerto serie RS-232 de la PC no significa que el protocolo utilizado para comunicar la PC y la placa del programador sea este, de hecho se puede adaptar este circuito para conectarlo al puerto paralelo e incluso USB. Como dijimos antes, los tiempos, y las señales necesarias para programar los PICs dependen de un protocolo específico desarrollado por Microchip, por lo que utilizamos el puerto como vehiculo para llevar los bits al PIC y para obtener las tensiones necesarias para la programación, pero utilizando un programa y un protocolo específico para esta tarea.

En las imágenes que acompañan este artículo encontraran el esquema eléctrico y el trazado

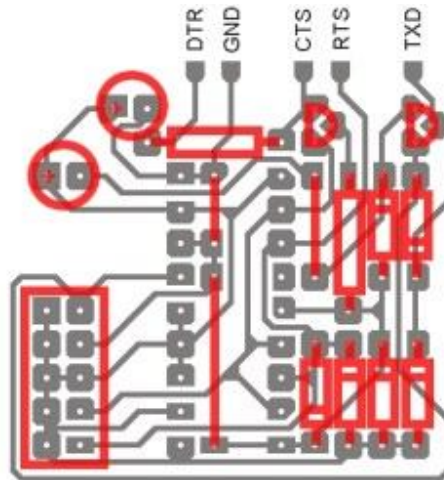
del circuito impreso necesario para montar todos los componentes. La forma de construir un circuito impreso de manera muy fácil y rápida, con resultados excelentes, está explicada al detalle en un artículo de NeoTeo.

Una vez que tengamos el circuito impreso listo, agujereado y bien limpio (libre de gratitud), procedemos a soldar los componentes. Son unos pocos y no debería haber problemas. Es preferible soldar primero los puentes, el zócalo para el PIC, y luego los demás componentes, teniendo cuidado de respetar el sentido en que colocamos los diodos, transistores y los condensadores, si no el programador no funcionara. Debemos decidir si usamos una ficha DB9 o DB25, y soldar un cable plano de al menos 5 hilos de aproximadamente un metro de largo entre la plaqueta y la ficha, por supuesto teniendo cuidado de no equivocarnos con la numeración de los pines (ver recuadro para conocer la disposición de pines de ambos conectores).

Como ocurre con cualquier proyecto de este tipo, y más si lo vamos a conectar a un puerto de nuestro ordenador, conviene revisar concienzudamente que todas las conexiones sean correctas, y que al soldar no hayamos hecho algún “puente” que pueda darnos dolores de cabeza.



- *Posición de los componentes en la plaqueta.*

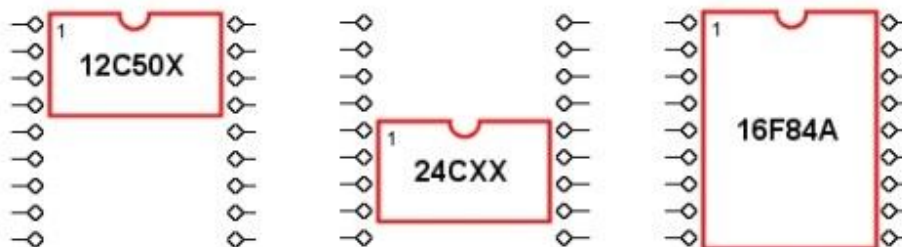


- *Circuito impreso.*

4 diodos 1N4148
 1 diodo Zener de 8.2v
 1 diodo Zener de 5.1v
 1 capacitor electrolítico de 100uF x 40v
 1 capacitor electrolítico de 22uF x 16v
 2 transistores BC547B
 1 resistencia de 10k, ¼ de watt
 1 resistencia de 1.5k, ¼ de watt.
 1 zócalo de 18 pines

Además necesitamos un trozo de cable plano de al menos 5 hilos, un conector DB-9 o DB-25 según sea el que tengamos en la PC, y opcionalmente un conector tipo molex de 8 pines para la programación “in circuit”.

El impreso lo construiremos como ya hemos visto en otras notas, o bien podemos usar un trozo de circuito impreso del que se emplea para construir prototipos, que ya tiene dibujadas las islas y los agujeros hechos, y solo tenemos que soldar puentes para “crear” nuestro diseño.



Posición de los chips en el zócalo.

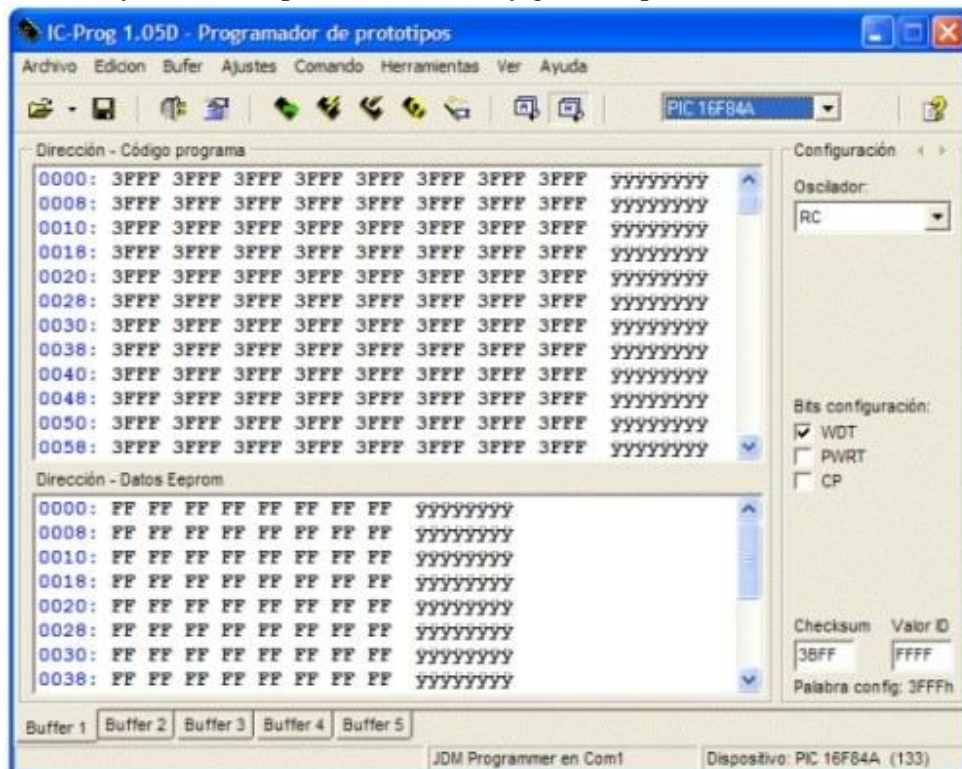
Es capaz de grabar y leer prácticamente todos los PICs y casi todos los tipos de memorias que hay en el mercado. Podemos bajarlo gratuitamente de la pagina de su autor (ver links) , de donde elegiremos la versión “IC-Prog Software 1.05D Multi-Lingual” y si estamos usando Windows NT o XP también necesitaremos descargar un driver especial, que también encontraremos en ese sitio y al que deberemos descomprimir en la misma carpeta donde hayamos puesto el IC-Prog.

En la misma página hay un excelente archivo de ayuda, en español, que deberíamos leer para usar sin problemas el programa.

Es importante seleccionar el programador JDM en la opción “Ajustes” -> “Tipo Hardware” ya que si no será imposible grabar nada. En dicha pantalla de opciones figura un control deslizante con el nombre “Retardo I/O”. Este valor es la demora entre cada dato enviado al PIC. Cuando mas pequeño, menos tardaremos en grabar el programa, pero de acuerdo a la PC que tengamos no podremos hacerlo demasiado pequeño. 10 es un buen valor, y se pueden ir haciendo pruebas con valores mas pequeños. Cuando veamos que la grabación falla, volvemos al valor que funciona y lo dejamos así.

No hay mucho mas para decir de este sencillo programa, y después de haber grabado un par de microcontroladores lo usaremos con confianza y seguridad. Por supuesto, también puede ser utilizado para el proceso inverso, es decir, leer el contenido de un microcontrolador.

Esto solo será posible si al momento de grabarlo no se utilizo la opción de proteger su contenido, en cuyo caso solo podrá ser borrado y grabado pero no leído.



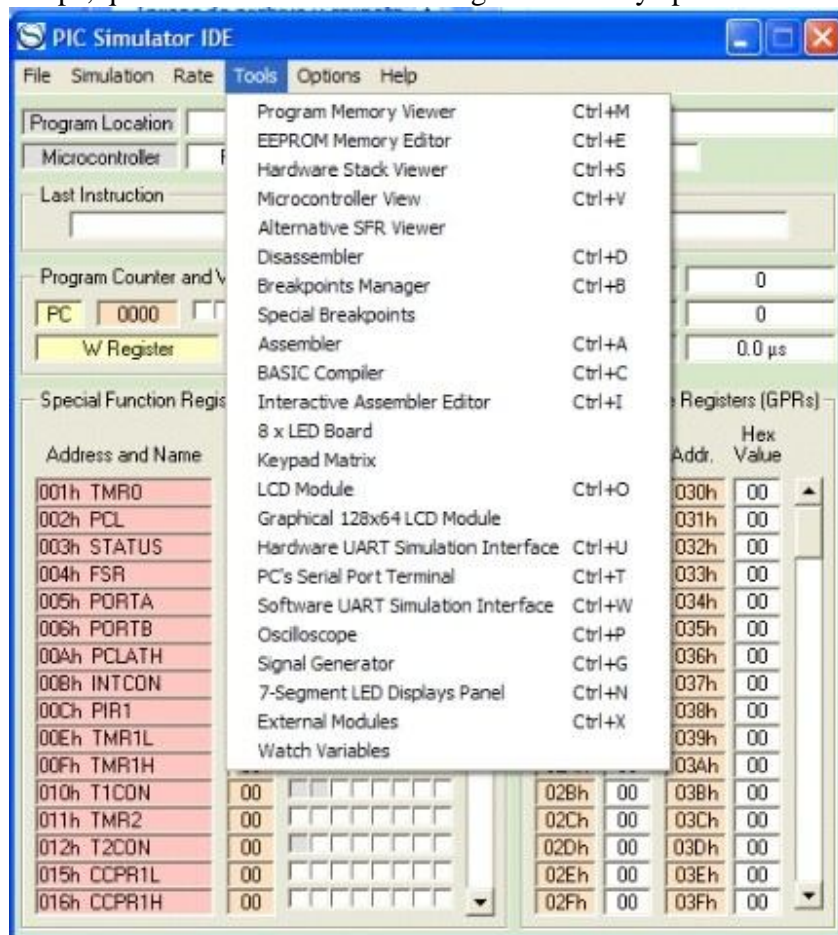
IC-PROG, el software de grabación.

El programa, desde el punto de vista del PIC es una serie más o menos larga de “0” y “1”. Como sería sumamente complicado programar de esa manera, se utiliza un ASM provisto por Microchip (u otros lenguajes de alto nivel, principalmente C y Basic) que al compilarse genera un archivo con extensión .HEX que es el que se carga con la ayuda del IC-PROG y el programador en el microcontrolador.

A lo largo de esta serie de notas iremos explicando como construir nuestros propios programas de alto nivel, y como convertirlos en ASM.

Utilizaremos una versión de BASIC, lo que significa que será muy fácil de aprender a sacar el jugo a estos chip, ya que es un lenguaje muy sencillo de aprender y probablemente muchos lectores en algún momento hayan hecho algún programa en su ordenador utilizando uno de los tantos dialectos de BASIC disponibles.

La idea es que desde NeoTeo brindemos lo necesario para que el aficionado aprenda a utilizar estos chips, que hacen de la electrónica digital un hobby apasionante.



PIC Simulator IDE, un BASIC para PIC.

Fuente: NeoTeo