

# Programadores para Microcontroladores PIC

*M. en C. Juan Carlos González Robles,  
Ing. Agustín Cruz Contreras,  
M. en C. Juan Carlos Herrera Lozada,  
Profesores del CIDETEC-IPN.*

**A**unque la mayoría de los PICs (Peripheral Interface Circuits, Circuitos de Interfase Periférica) utilizan esencialmente el mismo sistema de programación, existen entre ellos sutiles diferencias, no sólo en cuanto a la disposición de las terminales (pines de conexión) utilizadas para la programación, sino también en cuanto a las señales eléctricas necesarias, lo que hace que algunos programadores sólo puedan programar ciertos tipos de PICs. Aquí se presentan dos programadores con distintas posibilidades y limitaciones, para que el lector elija el que se adapte mejor a sus necesidades.

---

## ANTECEDENTES

---

Aun cuando no se pretende profundizar sobre los microcontroladores, es conveniente explicar qué son y cuáles son sus componentes. Un microcontrolador es un chip que, como mínimo, contiene en su interior un microprocesador, una memoria tipo ROM (que alberga el programa que ejecutará el microprocesador), una memoria RAM (para almacenar variables y datos), y unos puertos de entrada y/o salida que le permitirá

interactuar con otros dispositivos. Lo que diferencia a unos microcontroladores de otros son las características de cada uno de estos elementos y la inclusión o no de elementos adicionales, como memoria EEPROM, circuitería de reloj, temporizadores, contadores, convertidores AD, puertos serie, módulos PWM, etc. Cuando hablamos de programar un microcontrolador, básicamente nos referimos a grabar en la memoria de programa una serie de datos e instrucciones contenidos en un archivo, que corresponden al programa que ejecutará el microcontrolador durante su funcionamiento.

La utilización de microcontroladores en circuitos electrónicos comerciales se ha convertido en algo habitual desde hace unos años, debido principalmente a que permiten reducir el tamaño y el precio de los productos finales, pero hasta no hace mucho, resultaba poco menos que aterrador para el aficionado a la Electrónica incluirlos en sus montajes por diversas razones, como la dificultad para encontrarlos, su precio al por menor, la necesidad de complejos dispositivos para programarlos y lo arduo de escribir programas para ellos por la escasez y precio de herramientas software potentes. Además, la mayoría de los microcontroladores incorporaban una memoria de programa OTP (*One Time Programmable* – programable una sola vez), a la vez que los simuladores disponibles eran poco fiables, de forma que un

pequeño error en el código programado obligaba a desechar el microcontrolador.

En los últimos años, sin embargo, se ha abierto la vereda de los microcontroladores para los aficionados a la Electrónica, por varias razones. En primer lugar, el abaratamiento y el incremento de prestaciones han hecho que, aún siendo caros al por menor (un PIC16F84A cuesta industrialmente menos de 47.00 pesos, frente a los 80.00 que cuesta en los comercios), merezca la pena utilizarlos. Por otro lado, la política de varios fabricantes de ofrecer documentación y software de programación y simulación de forma gratuita, junto con cuestiones técnicas como la posibilidad de reprogramar los dispositivos (en muchos la memoria de programa es Flash, EPROM o EEPROM), y la simplicidad de los programadores necesarios (gracias a la implantación de protocolos de programación en serie frente a los antiguos sistemas de programación en paralelo), han hecho que hoy en día resulte muy fácil incluirlos en los diseños amateur.

Diversos fabricantes ofrecen gamas de microcontroladores para todos los gustos, aunque sin duda, el que se ha ganado el mercado para diseños sencillos (y buena parte de los profesionales) de los últimos tiempos es Arizona Microchip, con su conocida familia de microcontroladores PIC. Una de las razones es que todos utilizan el mismo juego de instruccio-

nes RISC (que además es muy reducido, son sólo 35 instrucciones distintas) y una filosofía de funcionamiento muy similares, de forma que si se aprende a utilizar un modelo concreto, es muy fácil adaptarse a otros, lo que resulta muy cómodo para el aficionado. Además, Microchip ofrece gratuitamente el editor y simulador de código **MPLAB**, bastante eficaz.

El objeto de este artículo es presentar una serie de programadores orientados especialmente a esta gama de microcontroladores. A lo largo del artículo se nombrarán diversos modelos de PICs. Para conocer a detalle sus características se recomienda visitar la página de Arizona Microchip, [www.microchip.com](http://www.microchip.com).

### DESARROLLO

Si bien son muchos los programadores disponibles en la Web que permiten programar microcontroladores PIC, se decidió tomar los más sencillos en cuanto a su implementación y elaboración (componentes económicos y fáciles de conseguir en nuestro país, software libre y fácil de usar y en español), y probarlos para así poder publicar el presente artículo o nota de aplicación de programadores fáciles y económicos para aficionados, estudiantes y docentes en el área de electrónica. La ventaja principal de hacerlo así es que son compatibles con gran parte del software de programación existente. Los dos programadores han sido probados con distintos PICs, lo que permite, en cada caso, enumerar sus posibilidades y limitaciones con bastante exactitud.

La razón de publicar los dos, es que el primero, si bien es limitado, su costo no pasa de 70.00 pesos y permite programar los modelos de PICs que se utilizan en la mayoría de los diseños prácticos, por lo que son idea-

les para quien necesite programar un PIC de forma constante, quedando el segundo, que tiene un costo que no pasa de 80.00 pesos, para los aficionados a la Electrónica que quieran hacer sus propios diseños y necesiten una herramienta flexible, segura y casi universal.

El primer programador es una versión modificada de uno de los muchos diseños de **JDM** (Diseñador Jens Dyekjaer Madsen). Tan sólo es capaz de programar los PICs 12C50x, 16x84, y las memorias EEPROM seriales 24Cxx. Utiliza el puerto serie para comunicarse con la PC, y de él obtiene los voltajes necesarios, por lo que no se necesita de una fuente de alimentación. La desventaja de este sistema es que si las señales del puerto serie no cumplen con la norma RS232, como ocurre en muchas portátiles, no funcionará correctamente. Sin embargo en la mayoría de las PCs de escritorio no habrá problema. Es compatible con el software IC-Prog (software libre para prototipos de programadores, el cual se encuentra en la dirección WEB: <http://www.ic-prog.com>), que permite realizar las operaciones necesarias (leer, borrar y grabar el PIC). Los detalles sobre este programador se encuentran descritos en el punto 3.1 Programador JDM2.

El segundo es una modificación del **Ludipipo** adaptada a la programación de diversos PICs. Al igual que el

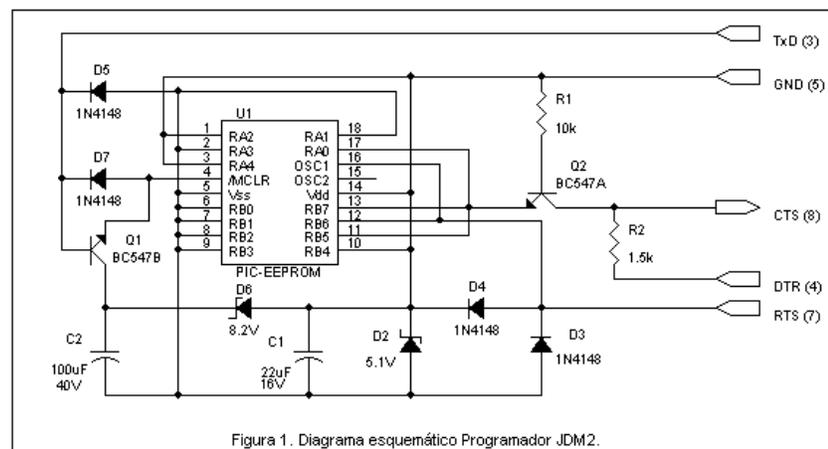
anterior, para comunicarse con el PC utiliza el puerto serie, y de él toma los voltajes necesarios. Tiene, por lo tanto, el mismo problema que el diseño anterior al utilizarlo con PCs portátiles. Es compatible con el software IC-Prog y permite programar varios modelos de PICs, incluidos 16F8x, 16F62x, 16F87x, 18Fxx2 y 18Fxx8. Los detalles se encuentran descritos en el punto 3.2 Programador PIPO2.

### PROGRAMADOR JDM2

#### EL CIRCUITO ESQUEMÁTICO

El programador que a continuación se presenta está dirigido a quien necesite programar alguno de los dispositivos mencionados con anterioridad. No es perfecto, falla en algunas PCs (portátiles sobre todo). Pero si todo lo que queremos es programar alguno de estos PICs, cuyo precio anda por los 80.00 pesos, tampoco estaría justificado gastar un dineral en un programador profesional (ya sea Microchip, Motorola, National Semiconductor, etc.) cuyo precio oscila de los 2,000.00 pesos en adelante. Este programador se puede construir sobre una placa fenólica universal (las placas perforadas para prototipos) de sólo 4 cm x 4.6 cm, en menos de una hora.

En la **figura 1**, se puede ver el circuito esquemático del programador. Sin profundizar en su funciona-



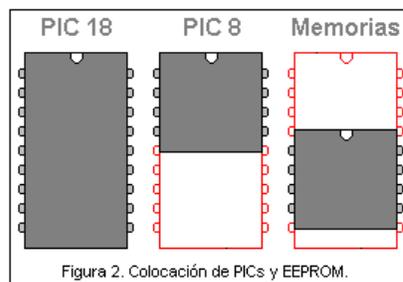
miento, para programar un PIC se necesitan dos voltajes respecto a tierra, +5V para la alimentación y +12.5V para la señal  $V_{pp}$  (voltaje de programación del PIC). La norma RS232 (puerto serie de la PC) establece que el voltaje correspondiente al 0 lógico debe estar entre -8V y -12V y el correspondiente al 1 lógico entre +8V y +12V. ¿Cómo conseguir entonces la señal  $V_{pp}$  necesaria para programar el PIC, que debe ser de +12.5V?

El truco está en crear una tierra para la alimentación del PIC que en realidad está a -5V respecto a la tierra del puerto serie. De esta forma, la tierra del puerto serie (pata 5 del conector DB9) queda a +5V respecto a la tierra del PIC (pata 5 o 18 de la base para cualquiera de los PICs), y de hecho, es la tierra del puerto serie la que se utiliza como alimentación de +5V para el PIC. El condensador C1 se carga mediante los diodos D3 y D4, y sirve de fuente de voltaje de +5V, que queda estabilizada mediante el diodo zener D2. De forma parecida, C2 se carga mediante D5 y Q1, quedando su voltaje estabilizado a unos +13V gracias a la unión en serie de los diodos zener D6 y D2. Este voltaje es el que se utiliza como voltaje de programación para el PIC.

Para las demás señales necesarias para la programación, DATA y CLOCK se utilizan directamente las señales del puerto serie CTS (DATA desde el PIC), DTR (DATA hacia el PIC) y RTS (CLOCK hacia el PIC).

Esta filosofía es la utilizada en la mayoría de programadores que toman su alimentación del puerto serie, pero la mayoría fallan a la hora de programar los PIC por las especiales características de este PIC en cuanto a temporización y consumo de  $V_{pp}$ . Este diseño, se probó con el software libre de desarrollo IC-Prog versiones 1.04c y 1.05a.

La colocación de los diferentes dispositivos a programar se muestra en la figura 2. Se dispone de un zócalo de 18 pines el cual permite conectar, tanto PICs como memorias EEPROM seriales. Los PIC's de 18 pines sólo se colocan ocupando la totalidad del zócalo mientras que los PIC's de 8 pines se colocan ocupando la parte superior del mismo. En tanto las memorias seriales se colocan una hilera anterior al fin del zócalo (el pin 1 de la memoria debe coincidir con el 5 del zócalo).



### PROGRAMADOR PIPO2 (LUDIPIPO)

#### EL CIRCUITO ESQUEMÁTICO

Este programador, basado en el Ludipipo, está adaptado a la programación de PICs, es simple y barato, pero permite programar a través del puerto serie de cualquier PC de escritorio (en las portátiles da problemas) una cantidad nada despreciable de PICs distintos, y todo ello utilizando software libre de programación estándar, como IC-Prog 1.05A.

Se pueden programar sin problemas los siguientes PICs:

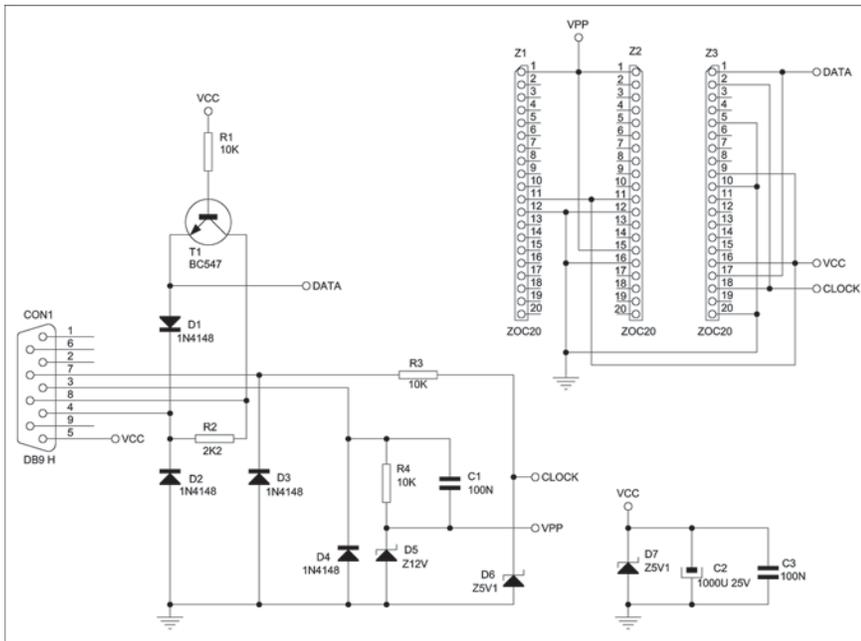
16F627, 16F628.  
16C84, 16F83, 16F84.  
16F873, 16F874, 16F876, 16F877.  
18F242, 18F252, 18F258, 18F442,  
18F452, 18F458.

Y, al menos en teoría (no se han probado) debería funcionar correctamente con los siguientes:  
16C62, 16C63, 16C64, 16C65,  
16C66, 16C67.

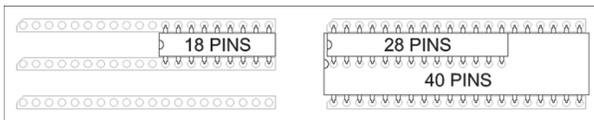
16C71, 16C72, 16C73, 16C74,  
16C75, 16C76, 16C77, 16C715.  
16C620, 16C621, 16C622,  
16C623, 16C624, 16C625.  
16F870, 16F871, 16F872.  
16C923, 16C924.

Sin entrar en detalles sobre el funcionamiento del circuito esquemático de la **figura 3**, diremos sólo que se basa en principios muy parecidos a los del JDM, pero con ciertos retoques en la temporización y la estabilidad de las señales. En este caso, se utiliza como alimentación de +5V el condensador C2, que se carga mediante D2, D3 y D4 en los momentos en que las señales DTR, RTS y TxD del puerto serie son negativas, y su voltaje queda estabilizado mediante D7. D5 fija la tensión de programación en +12V y D6 limita la tensión de la señal CLOCK a +5V.

La colocación de los diferentes dispositivos a programar se muestra en la **figura 4**. Se dispone de un zócalo compuesto de tres tiras de pines el cual permite conectar, tanto PIC's de 40, 28 y 18 pines. Los PIC's de 40 pines se colocan ocupando la totalidad del zócalo (quedando libre la tira de en medio debajo del PIC) mientras que los PIC's de 28 pines se colocan ocupando la parte superior del zócalo (quedando el pin 1 sobre la tira de en medio, y el pin 28 en lo que sería el pin 40 del de 40), en tanto que los de 18 pines quedan montados también sobre la tira de en medio, pero sobre la parte inferior quedando el pin 9 en el lugar 20 de esta tira, y el pin 10 en lo que sería el pin 21 del PIC de 40.



**Figura 3. Diagrama esquemático del Programador PIPO2.**



**Figura 4. Colocación de PICs en sus diferentes tamaños.**

## CONECTOR

Para conectar el programador a la PC hará falta también un cable serie transparente (cableado pin a pin) que tenga cableados al menos los pines que se indican en la **figura 5**. Este cable se puede fabricar o comprarlo ya hecho (es el tipo de cable que se utiliza para conectar un modem a la PC).

## CONCLUSIONES

Lo primero que hay que tener claro a la hora de utilizar estos programadores es el orden en que se deben hacer las cosas para no estropear ni el programador, ni el PIC, ni el puerto serie de la PC. Siempre que queramos insertar o extraer un PIC del zócalo hay que desconectar el programa-

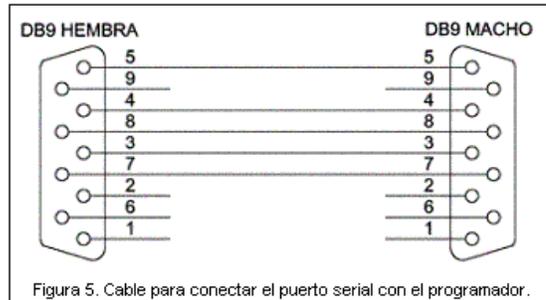
del puerto serie, ya que, al extraer la alimentación del puerto serie, mientras esté conectado estará alimentado. Por lo tanto, el proceso a seguir consta de los siguientes pasos:

- 1º. Con el programador desconectado insertar el PIC en el zócalo en la posición correcta (figuras 2 o 4).
- 2º. Conectar el programador al cable que viene del puerto serie de la PC.
- 3º. Llevar a cabo las operaciones de grabación o lectura necesarias.
- 4º. Desconectar el programador del cable que viene del puerto serie de la PC.
- 5º. Extraer el PIC del zócalo.

Por otro lado, es necesario instalar y configurar adecuadamente el software para poder interactuar con el programador (JDM2 o PIPO2). Esta instalación se detalla en el artículo "IC-Prog software para prototipos de Programadores".

## BIBLIOGRAFÍA E INTERNET

- [1] José M<sup>a</sup>. Angulo Usategui, Ignacio Angulo Martínez. "Microcontroladores PIC: Diseño práctico de aplicaciones". Editorial Mc Graw Hill, 2<sup>a</sup>. Edición, 1999.
- [2] José M<sup>a</sup>. Angulo Usategui, Susana Romero Yesa, Ignacio Angulo Martínez. "Microcontroladores PIC: Diseño práctico de aplicaciones. Segunda Parte: PIC16F87X". Editorial Mc Graw Hill, 2<sup>a</sup>. Edición, 2000.
- [i] [www.microchip.com](http://www.microchip.com)
- [ii] [www.ic-prog.com](http://www.ic-prog.com)
- [iii] [www.jdm.homepage.dk/newpic.htm](http://www.jdm.homepage.dk/newpic.htm)
- [iv] [club.telepolis.com/vayas10/programadores.htm](http://club.telepolis.com/vayas10/programadores.htm)
- [v] [213.97.130.124/progs/progs.htm](http://213.97.130.124/progs/progs.htm)
- [vi] [www.redeya.com](http://www.redeya.com)
- [vii] [www.geocities.com/pictarjetas/esquema3.htm](http://www.geocities.com/pictarjetas/esquema3.htm)
- [viii] [www.msebilbao.com/tienda/default.php](http://www.msebilbao.com/tienda/default.php)
- [ix] [skyscraper.fortunecity.com/email/250/pic.htm](http://skyscraper.fortunecity.com/email/250/pic.htm)
- [x] [www.pic-compiler.com/hw/p\\_piccolog.htm](http://www.pic-compiler.com/hw/p_piccolog.htm)
- [xi] [www.blichfeldt.dk/propic/](http://www.blichfeldt.dk/propic/)



**Figura 5. Cable para conectar el puerto serial con el programador.**