

Protocolo de programación de dispositivos Microchip PIC: In Circuit Serial Program (ICSP™)

Bravo Muñoz, O. Durán Mora, J.L. Lázaro Galilea

Departamento de
Electrónica, Universidad
de Alcalá
ibravo@depeca.uah.es;
http://
www.depeca.uah.es

En el mundo de la electrónica, como en cualquier otro, existe una gran competencia entre fabricantes de dispositivos. En este caso nos encontramos ante el problema de elegir un microcontrolador para desarrollar un proyecto. ¿Cuál escoger?

Una de las características que pueden inclinar la balanza hacia uno u otro fabricante, (independientemente del precio por unidad) entre las muchas que poseen, es la facilidad de programación. Es decir, que el dispositivo presente un sistema de programación sencillo y rápido que minimice los tiempos de desarrollo software. Este es el caso de Microchip® y sus dispositivos PIC® que presentan un protocolo de programación serie, que utiliza un número de señales mínimas y además es programable en circuito pudiendo actualizar el software en la propia tarjeta. A este protocolo de programación se le denomina ICSP o programación serie en circuito.

Este artículo forma parte de una trilogía, dedicada a presentar un sistema software hardware, encargado de programar y manejar los microcontroladores PIC de gama media / baja. Se presentan los fundamentos teóricos del protocolo de programación de estos dispositivos. Los conceptos desarrollados en éste son necesarios a la hora de abordar con éxito la creación de plataformas con dispositivos de tipo PIC.

Introducción al ICSP "In Circuit Serial Programming"

El ICSP, es el sistema utilizado en los dispositivos PIC de Microchip para programarlos. Es válida para todos los PIC de la gama baja como puede ser el 12C508 y 12C509, los de la gama media, desde el de 16F84 hasta el 16F877 e incluso para las nuevas familias 18xxxx.

Se trata de un sistema de programación serie síncrona en el que

intervienen 2 señales, una de entrada y salida para la transmisión y recepción de datos y otra de entrada para la sincronización de la transmisión y recepción de los datos. Las líneas utilizadas se ubican en el pin RB6/PGC para la señal de sincronización (reloj) y en el RB7/PGD para los datos.

El protocolo trabaja con dos tensiones, una de alimentación (VDD), cuyo rango de valores está comprendido entre 4.5 y 5.5 voltios, y otra de programación (VPP) cuyo rango oscila entre un mínimo y un máximo de 12 y 14 voltios respectivamente.

En la Tabla 1 aparece una descripción de los pines necesarios para llevar a cabo una programación serie.

Programación del dispositivo

Los PIC integran en el chip inter-

no un protocolo de programación. Para accionar este modo programación la tensión del pin MCLR debe variar su valor desde VIL (0 voltios) hasta el valor de la tensión de programación $V_{IH}(V_{pp})$. El chip entraría entonces en un modo denominado por Microchip "Program/Verify". Una vez en este modo se puede acceder a la memoria de programa, de datos y a la de configuración para programarla y verificarla. En este modo RB6 y RB7 son entradas *Trigger Schmitt*. El tiempo para pasar del nivel VIL al VIH debe ser inferior a 72 ciclos de reloj para asegurar que el dispositivo a entrado en este modo.

La comunicación con el chip se realizará mediante una serie de comandos específicos, que aparecen en la Tabla 2 y la Tabla 3, y mediante una transmisión serie se podrá acceder a leer, borrar y programar cada uno de los bytes de memoria del dispositivo. Los comandos presentan un valor

Tabla 1. Descripción de los pines utilizados en la programación ICSP.

Pin	Función	Tipo de pin	Descripción
RB6	CLOCK	Entrada	Entrada de reloj
RB7	DATA	Entrada / salida	Entrada y Salida de Datos
MCLR	VPP	Programación	Tensión de Programación
VDD	VDD	Alimentación	Tensión Positiva
VSS	VSS	Alimentación	Tensión Negativa

Tabla 2. Comandos para la programación ICSP los dispositivos PIC's 16F83/CR83/F84/CR84.

Comandos	MSB . . . LSB	Dato
Cargar configuración	0 0 0 0 0 0	0, dato(14 bits),0
Cargar dato en memoria de programa	0 0 0 0 1 0	0, dato(14 bits),0
Leer dato de memoria de programa	0 0 0 1 0 0	0, dato(14 bits),0
Incrementar dirección	0 0 0 1 1 0	
Inicio de programación	0 0 1 0 0 0	
Cargar datos en memoria de datos	0 0 0 0 1 1	0, dato(14 bits),0
Leer dato de memoria de datos	0 0 0 1 0 1	0, dato(14 bits),0
Borrar memoria de programa	0 0 1 0 0 1	
Borrar memoria de datos	0 0 1 0 1 1	

Tabla 3. Comandos para la programación ICSP para los dispositivos PIC's 16F84A/16F877.

Comandos	MSB . . . LSB	Dato
Cargar configuración	x x 0 0 0 0	0, dato(14 bits),0
Cargar dato en memoria de programa	x x 0 0 1 0	0, dato(14 bits),0
Leer dato de memoria de programa	x x 0 1 0 0	0, dato(14 bits),0
Incrementar dirección	x x 0 1 1 0	
Inicio de programación con borrado	0 0 1 0 0 0	
Inicio de programación	0 1 1 0 0 0	
Cargar datos en memoria de datos	x x 0 0 1 1	0, dato(14 bits),0
Leer dato de memoria de datos	x x 0 1 0 1	0, dato(14 bits),0
Borrar memoria de programa	x x 1 0 0 1	
Borrar memoria de datos	x x 1 0 1 1	

distinto para que la circuitería interna del chip pueda discriminar uno de otro. Además, no todos los dispositivos pueden utilizar todos los comandos.

La transmisión de un comando se realiza en 6 ciclos de reloj. Cada bit de un comando se registra durante el flanco de bajada del pulso de reloj siendo el bit menos significativo (LSB) el primero en enviarse. Los datos deben respetar un tiempo de *setup* y de *hold* (ver especificaciones AC/DC de cada dispositivo) anterior al flanco de bajada del pulso de reloj.

Existen comandos de control para realizar las comunicaciones y comandos de datos. Entre los comandos que trabajan con datos (carga y lectura) se debe respetar un tiempo de espera mínimo de 1 μ s entre el comando y el dato. Una vez transcurrido este tiempo el dato se envía en 16 ciclos de reloj enviándose primero el bit menos significativo. En una operación de escritura de datos el bit es registrado en el flanco de bajada del segundo ciclo de reloj (ver gráfica de la figura 1).

Durante la operación de lectura dicho bit es transmitido por el PIC en el flanco de subida del segundo ciclo de reloj (ver gráfica figura 2).

La escritura de un dato en memoria se realiza con un ciclo de programación formado por un comando de carga de, un comando de inicio de programación y una espera de 10 ms (figura 3) para que la escritura del dato tenga lugar con éxito. Este tiempo de espera va a limitar la velocidad de programación siendo nunca mayor de 100 palabras por segundo. Para llegar a una posición de memoria concreta se utiliza el comando de incremento de dirección.

Comandos de la programación ICSP

De los comandos disponibles por este tipo de microcontroladores,

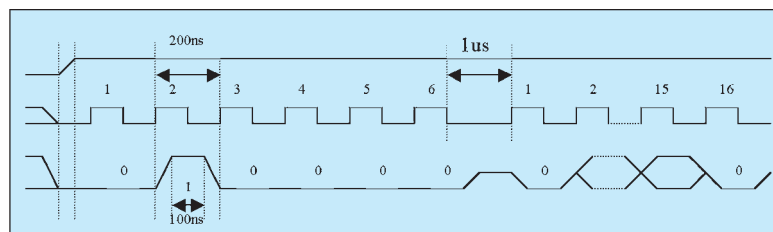


Figura 1. Cronograma escritura de datos.

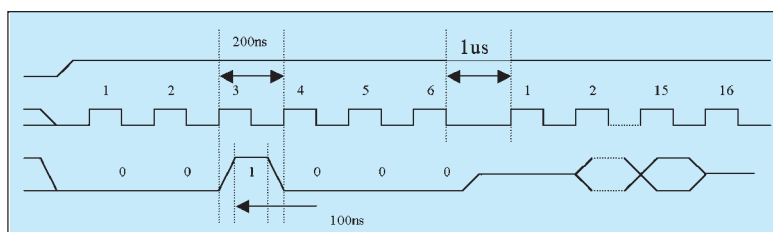
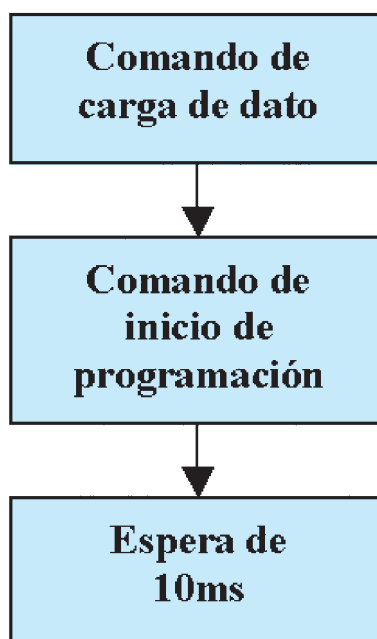


Figura 2. Cronograma de la lectura de datos



• Carga de la configuración

El PIC al recibir este comando situará el puntero de programación en la posición 0x2000. A partir de esta dirección y hasta la 0x2007 aparecen los registros de configuración como se indica en la Tabla 4. Los registros de identificación del dispositivo, *ID Locs*, serán los primeros que deberán cargarse. En estos registros se puede introducir un código de 4 bytes que no referencie al dispositivo. A continuación se cargará la palabra de configuración que regirá el modo de funcionamiento del chip.

Para salir de esta situación y poder situar el puntero en otra dirección de memoria, ya sea de datos o de programa, es necesario reiniciar el

Figura 3. La escritura de un dato en memoria se realiza con un ciclo de programación formado por un comando de carga de, un comando de inicio de programación y una espera de 10 ms

Tabla 4. Registros de configuración.

Dirección	
0x2000	Localización ID
0x2001	Localización ID
0x2002	Localización ID
0x2003	Localización ID
0x2004	Reservado
0x2005	Reservado
0x2006	Reservado
0x2007	Palabra de configuración

los autores han implementado aplicaciones así como tarjetas de programación y depuración, en base al protocolo ICSP. De todo el conjunto de comandos del que se dispone, han sido implementados los que aparecen en la Tabla 2 y Tabla 3 y son los siguientes:

chip estableciendo en la patilla MCLR una tensión igual a VIL.

- *Incremento de dirección*

Este comando se utiliza para controlar la dirección de memoria a la que se está accediendo, es decir, la posición del puntero. Después de entrar en modo programación el puntero presenta el valor de 0x0000.

Con la ejecución de este comando el puntero de programación se incrementará en una unidad ya sea para la lectura o la escritura de memoria. Para acceder a una posición de memoria concreta debemos enviar este comando tantas veces como posición tenga dicha memoria.

- *Escritura de datos en la memoria de programa*

Una vez recibido este comando el chip leerá una palabra de 14 bits durante la aplicación de 16 ciclos de reloj y la almacenará en la posición de memoria de programa en la que esté situado el puntero.

- *Escritura de datos en la memoria de datos*

Después de recibir este comando el chip cargará una palabra de 14 bits en la memoria de datos durante la aplicación de 16 ciclos de reloj siguiendo el diagrama de tiempos similar al comando "Lectura de datos de memoria de programa" y siendo el bit menos significativo el primero en recibir.

Sin embargo la memoria de datos tiene un ancho de 8 bits por lo que solamente los 8 primeros bits de los 14 recibidos serán almacenados en la memoria. Esto es debido a la necesidad de mantener los 16 ciclos de reloj para mantener la misma secuencia que con el acceso a memoria de programa.

- *Lectura de datos de la memoria de programa*

Una vez recibido este comando el chip transmitirá una palabra de 14 bits de la memoria de programa, ya sea memoria de usuario o de confi-

guración, apuntada por el puntero. El pin RB7 pasará a ser de salida en el segundo flanco de subida de la señal de reloj y volverá a alta impedancia en el flanco de subida del ciclo 16.

- *Lectura de datos de la memoria de datos*

Después de recibir este comando el chip transmitirá un dato de 14 bits empezando en el flanco de subida del segundo ciclo de reloj. Como en todo los comandos que trabajan con datos el proceso se realizará en 16 ciclos de reloj. Debido a que el ancho de la memoria de datos es de 8 bits sólo los 8 primeros serán significativos. El resto se complementará para provocar un correcto reset del comando.

- *Inicio de borrado*

Un mecanismo interno ejecutará el borrado de la memoria con la carga de este comando. Antes de la programación de la memoria apropiada deberá ejecutarse este comando para una correcta escritura.

- *Inicio de programación*

Para la programación de la memoria de programa o de datos el chip necesita recibir y decodificar este comando para una correcta escritura del dato. Para que el ciclo de programación se complete será necesario enviar este comando después de enviar el de carga de datos y esperar 10 ms para que finalice.

Este comando es equivalente a inicio de borrado excepto en que borra el contenido de la memoria antes de comenzar la escritura del dato para asegurar la programación. Se recomienda ejecutar un borrado de toda la memoria si utilizamos este comando.

- *Borrado de la memoria de programa*

Después de llevarse a cabo este comando el siguiente comando de

programación borrará entera la memoria de programa. Para una correcta ejecución de este comando se deberá seguir la siguiente secuencia:

1. Cargar un dato cuyo contenido sea todos unos.
2. Ejecutar el comando "Borrado de la memoria de programa".
3. Ejecutar el comando "Inicio de programación".
4. Esperar 10 ms para que se complete el borrado.

Si el contador de programa está apuntando a la memoria de test (0x2000- 0x200F) la memoria de programa de usuario y la memoria de programa de test serán borradas. La palabra de configuración no se borrará aunque el contador de programa este en al dirección 0x2007.

- *Borrado de la memoria de datos*

Después de ejecutar este comando si se envía un comando de programación de memoria de datos, la memoria de datos del chip se borrará entera. Para una correcta ejecución de este comando se deberá seguir la siguiente secuencia:

1. Cargar un dato cuyo contenido sea todos unos.
2. Ejecutar el comando "Borrado de la memoria de datos".
3. Ejecutar el comando "Inicio de programación".
4. Esperar 10 ms para que se complete el borrado.

Protección del código

Los dispositivos de Microchip pueden habilitar o deshabilitar la protección del código de programa, de manera que si intentamos leer el chip su resultado será como si todos sus valores estuviesen a 0. La posibilidad de leer y escribir la palabra de configuración y los identificadores cuando el código está protegido depende del dispositivo del que se trate.

Para habilitar la protección del código se utiliza la palabra de configuración. Para deshabilitar dicha pro-

6. Esperar 10 ms
7. Ejecutar el comando "000001"
8. Ejecutar el comando "000111"

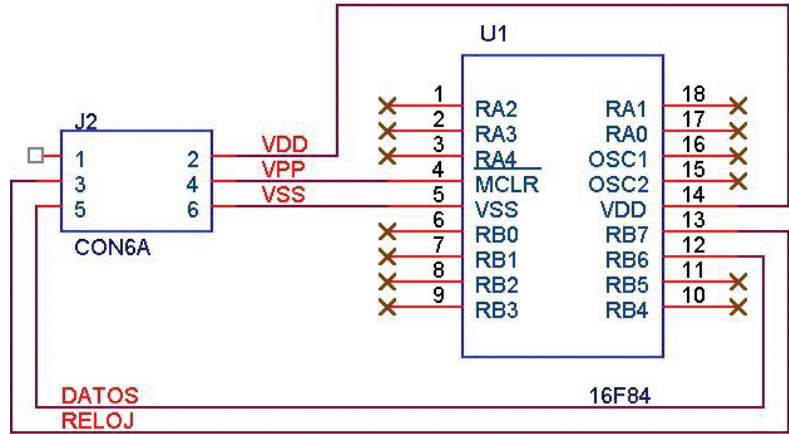


Figura 4. Líneas que intervienen en la programación

Conclusiones

El único inconveniente, que complica el desarrollo de aplicaciones de programación de estos chips, es el acceso a las posiciones de memoria. El sistema posee un único comando de incremento de posición por lo que si queremos acceder a un byte de memoria situado en la última posición debemos recorrer toda la memoria hasta llegar a esa posición. Mejoraría el sistema un coman-

Con la base teórica descrita en este artículo sobre el protocolo de programación de una de las clases de microcontroladores más vendidos actualmente, se abren las puertas para que cualquier diseñador pueda hacerse su propio sistema de programación de este tipo de dispositivos de una manera muy barata. Junto a este tipo de información que Microchip ofrece gratuitamente a través de su página web (<http://www.microchip.com>), también esta empresa distribuye un entorno de ensamblado de programas también de manera gratuita (MPLAB). El conjunto formado por estas dos características hace que cada día los usuarios de este tipo de dispositivos aumente notablemente, no cerrán-

Los autores del artículo han desarrollado un entorno de programación así como una placa de evaluación docente en base a estas características, que se pueden descargar gratuitamente de la dirección web de los mismos, cuyas características se podrán observar en artículos posteriores. □

Referencias

- Nota:** PIC®, Microchip®, ICSP™ son marcas registradas por Microchip Technology Incorporated.