

Protocolo ICSP: comandos

Load Configuration (00h)

- **Descripción:** Acceso a la memoria de configuración. Una vez ejecutado el contador de programa toma el valor 2000h. **Para volver a la memoria de programa es necesario hacer un reset.** En la memoria de configuración se encuentra la identificación del PIC y la palabra de configuración, valores que dependen del tipo de PIC empleado.
 - **Datos:** Para que se ejecute este comando hay que enviar un dato, aunque será ignorado (da igual su valor).
 - **Ejemplo:** Hacer reset, enviar el comando 00h y enviar el dato 00h. Ya nos encontramos en la memoria de configuración (pc=2000h). Ahora podemos utilizar otros comandos para movernos por ella así como leer y modificar los valores de sus posiciones
-

Load Data for Program Memory (02h)

- **Descripción:** Enviar un dato para su almacenamiento en la dirección actual del contador de programa. El dato sólo se envía, pero no se graba (se deja almacenado internamente hasta que se mande el comando de grabación. La forma de grabar depende del tipo de PIC).
 - **Datos:** Después de enviar el comando hay que transmitir el dato
 - **Ejemplo:** Enviar comando 02h y luego un dato de 14 bits. El dato quedará almacenado en un registro interno del PIC
-

Read Data from Program Memory (04h)

- **Descripción:** Lectura del dato que se encuentra en la dirección actual del PC. Se puede leer tanto información de la memoria de programa como de la memoria de configuración.
 - **Datos:** Después de enviar el comando el PIC transmite el dato que se encuentra en esa posición de memoria
 - **Ejemplo:** Enviar comando 04h y luego leer el dato de 14 bits.
-

Increment Address (06h)

- **Descripción:** Incrementar el contador de programa ($PC=PC+1$)
 - **Datos:** No hay transferencia de datos
 - **Ejemplo:** Enviar el comando 06h
-

Begin Erase/Programming Cycle (08h)

- **Descripción:** Grabación en la memoria de los datos previamente enviados (con el comando 02h). La forma de realizar la grabación depende del tipo de PIC.
 - **Familia 16F8X y 16F876:** Se envía un dato con el comando 02h y luego se realiza la programación (programación byte a byte)
 - **Familia PIC16F87XA:** Se envían 8 datos con el comando 02h y luego se realiza la programación (Programación de 8 en 8 bytes)
 - **Datos:** No hay transferencia de datos. **La grabación tarda 4ms en realizarse**
 - **Ejemplo:**
 - **Familia 16F8X y 16F876:** Comando 02h, enviar dato a grabar y por último el comando 08h
 - **Familia 16F87XA:** comando 02h, dato1, cmd 06h, cmd 02h, dato2, cmd 06h, cmd 02h, dato3, cmd 06h,..., cmd 02h, dato8 y por último el comando 08h. Esto graba los 8 datos enviados en las 8 direcciones de memoria consecutivas
-

Bulk Erase Program Memory (09h)

- **Descripción:** Borrar todo el contenido de la memoria de programa. Si el contador de programa está apuntando a la memoria de configuración (2000h-200Fh), ésta también se borrará (además de la de programa). **La palabra de configuración no se borra.** Esta operación depende del microcontrolador PIC, pero en la mayoría de los casos hay que enviar primero el valor 3fff usando el comando 02h. Para que se realice el borrado hay que enviar el comando "Begin Erase/program cycle (08h)".
 - **Datos:** No hay transferencia de datos. El borrado tarda unos 10ms en realizarse
 - **Ejemplo:** Para las familias 16F87X y 16F8X el borrado se hace de la siguiente manera. comando 02, dato 3fffh, comando 09h, comando 08h.
 - **Notas:** Comando **NO PROBADO** [1]
-

Bulk Erase Data Memory (0Bh)

- **Descripción:** Borrar toda la memoria de datos. Una vez enviado el comando es necesario enviar el comando 08h (Begin Erase/program cycle)
 - **Datos:** No hay transferencia de datos. El borrado tarda unos 10ms.
 - **Ejemplo:** Enviar comando 0Bh seguido por el comando 08h.
 - **Notas:** Comando **NO PROBADO** [1]
-

Load Data for Data Memory (03H)

- **Descripción:** Enviar un dato para su almacenamiento en la memoria de datos, direccionada por los bits menos significativos del PC (Esto depende del tipo de PIC). Por ejemplo, para el caso del PIC16F876A, que tiene 256 bytes de eeprom, se utilizan los 8 bits menos significativos del PC para acceder a esta memoria.
La memoria de datos es de 8 bits de ancho, pero la escritura se hace igual que si se estuviese

escribiendo en la memoria de programa, por lo que será necesario aplicar 16 pulsos de reloj (14 para los datos + 1 bit de start + 1 bit de stop). Sólo los primeros 8 bits enviados, después del bit de start, son los correctos. El resto son bits "basura" que será ignorados por el PIC.

- **Datos:** Después de transmitir este comando hay que enviar el dato.i>
 - **Ejemplo:** Este comando **NO SE HA PROBADO**. No hay suficiente información en las hojas de datos de Microchip. Es previsible que después de enviar este comando haya que ejecutar el comando 08h.
 - **Notas:** Comando **NO PROBADO** [1]
-

Read Data for Data Memory (05H)

- **Descripción:** Leer un dato de la memoria de datos. Esta memoria es de 8 bits de ancho, pero la lectura se hace igual que si se estuviese leyendo la memoria de programa, por lo que será necesario aplicar 16 pulsos de reloj (14 para los datos + 1 bit de start + 1 bit de stop). Sólo los primeros 8 bits recibidos, después del bit de start son los correctos. El resto es "basura" que se puede eliminar. Dependiendo del modelo de PIC, la memoria EEPROM tendrá un tamaño u otro. La dirección de acceso se toma de los bits menos significativos del contador de programa. Por ejemplo, para el caso del PIC16F876A, que tiene 256 bytes de eeprom, se utilizan los 8 bits menos significativos del PC para acceder a esta memoria.
 - **Datos:** Despues de enviar el comando ,el PIC transmite el dato que se encuentra en esa posición de memoria.
 - **Ejemplo:** Leer un dato de la primera posición de memoria de la eeprom: Hacer un reset, con lo que el PC=0000H. Enviar el comando 05h. Leer el dato de 14 bits. Sólo los 8 primeros bits son los correctos (Recordad que primero se envían los bits menos significativos)
 - **Notas:** Comando **NO PROBADO** [1]
-

Notas

[1] Comando no probado. A la fecha de escribir esta documentación (29/Dic/2003) ese comando no se ha probado en ningún PIC. El resto de comandos han sido implementados con éxito.

[Grabación de microcontroladores PIC]

IEA ROBOTICS

Juan González